دانشگاه صنعتي امير كبير

دانشكده مهندسي صنايع

رساله دوره دكتري (Ph.D) **مهندسي صنايع**

عنوان رساله :

طراحي سلولهاي توليدي در يک محيط چند معياري

Manufacturing Cell Design in a Multi-Criterion Environment

استاد راهنما :

آقاي دكتر سيد محمد معطر حسيني

دانشكده مهندسي صنايع، دانشگاه صنعتي امير كبير

اساتيد مشاور :

آقاي دكتر سيد حسن قدسي پور

(دانشكده مهندسي صنايع، دانشگاه صنعتي اميركبير)

آقاي دكتر سيد حسام الدين ذگردي

(دانشكده فني، دانشگاه تربيت مدرس)

تهيه كننده :

سيد افشين منصوري

بهمن ۱۳۷۹

بسم الله الرحمن الرحيم

چكيده به فارسي

چكيده: مسئله طراحي سلول هاي توليدي از سالهاي دهه ۷۰ ميلادي يكي از زمينه هاي تحقيقاتي در مبحث توليد سلولي بوده است. پيش از دهه ۹۰ اين مسئله عمدتا" با در نظر گرفتن يك معيار نظير: كمينه سازي حركات بين سلولي قطعات، بيشينه سازي شباهت ميان قطعات و/ يا ماشين ها در سلول ها و كمينه سازي عدم تعادل بار كاري سلول ها مورد بررسي قرار مي گرفته است. در سالهاي دهه ۹۰ و پس از آن مقالاتي كه مسئله طراحي سلول را به عنوان يك مسئله تصميم گيري چند معياره (MCDM) ، به ويژه در قالب يك مسئله بهينه سازي چند هدفي (MOP) مدل كرده اند به چاپ رسيده است.

در اين رساله يك مدل چند هدفي جديد براي تصميم گيري در مورد چگونگي برخورد با عناصر استثنايي در طراحي يك سيستم توليد سلولي ارائه شده است. مجموعه اهداف مورد نظر شامل : كمينه سازي حركات بين سلولي قطعات، كمينه سازي مجموع هزينه هاي تكرار ماشين ها و مقاطعه فرعي قطعات، كمينه سازي عدم بهره گيري از ماشين آلات و كمينه سازي عدم تعادل بار كاري سلول ها مي باشد. به دليل وجود تعارض ميان اين اهداف، دستيابي به يك جواب ايده آل، يعني جوابي كه به طور همزمان تمامي اهداف را بهينه نمايد، ممكن نيست. با اين وجود مي توان مجموعه اي از جواب هاي نامغلوب (بهينه پارتو) را جستجو نمود كه تصميم گير بتواند با توجه به ترجيحات خود، از ميان آنها انتخاب نمايد. بدين منظور يك الگوريتم ژنتيك چند هدفي بنام XGA توسعه يافته و به زبان Visual C++ كد شده است. اين الگوريتم از ايده سورت نامغلوب براي رتبه بندي افراد در جامعه بهره مي گيرد. توسعه اي از طرح انتخاب RSSWP كه در آن از يك طرح جديد نخبه گراي تصادفي استفاده شده، به كار رفته است. از آشيانه سازي براي حفظ پراكندگي جواب ها در مجموعه نخبه ها استفاده شده است. معيار توقف الگوريتم به گونه اي طراحي شده كه همگرايي الگوريتم به سوي لبه بهينه پارتو را همراه با حداكثر تعداد نسل ها در نظر مي گيرد.

تعدادي مسئله طراحي سلول مطرح در ادبيات توسط الگوريتم XGA و نيز سه الگوريتم مبنا به نامهاي: VEGA ، NPGA و NSGA حل شدند. نتايج بدست آمده بيانگر بهبود قابل ملاحظه در سه بعد : كيفيت، پراكندگي و زمان CPU در مقايسه با الگوريتم هاي مبنا بوده است.

واژه هاي كليدي :

مسئله طراحي سلول

مسئله تشكيل سلول

سيستم توليد سلولي

تصميم گيري چند معياره

بهينه سازي چند هدفي

الگوريتم هاي ژنتيك

چکیده به انگلیسي

Abstract: Manufacturing Cell Design problem has been a research area in the cellular manufacturing context since 1970's. Prior to the 1990's, the problem had been examined basically with respect to only a single criterion, e.g.: minimizing intercellular parts movements, maximizing similarity among the parts and/or machines in the cells, minimizing imbalance of the work load among the cells. In the 1990's and after, a number of papers have been published that model the cell design problem as a Multi-Criterion Decision Making (MCDM) problem, in particular as a Multiobjective Optimization Problem (MOP).

In this thesis a new multiobjective model has been developed to deal with the exceptional elements in the design of a Cellular Manufacturing System. The set of objectives include: minimizing intercellular part movements, minimizing sum of machine duplication and part subcontracting cost, minimizing machinery underutilization and minimizing imbalance of the cells' workloads. Due to the conflicts among the objectives, attaining an ideal solution, i.e. a solution that simultaneously optimizes all of the objectives, is impossible. However a set of non-dominated (or Pareto Optimal) solutions could be sought from which the decision maker will be able to select based on his or her priorities. To this end, a multiobjective genetic algorithm called XGA was developed and coded in Visual C++. The algorithm makes use of the non-dominated sorting idea to rank individuals in the population. An extension of *Reminder Stochastic Sampling Without Replacement* (RSSWP) is developed wherein a novel probabilistic scheme of elitism is applied. Niching is employed to keep diversity among the individuals in the elite set. Stopping criteria of the algorithm is devised so that takes into account convergence of the algorithm to the Pareto-Optimal frontier along with the maximum number of generations.

A number of cell design problems taken from the literature were solved by the XGA and also by three reference algorithms, namely: VEGA, NPGA and NSGA. The results obtained by the XGA show promising improvements in three dimensions, i.e. quality, diversity and CPU time compared with the reference algorithms.

KEY WORDS:

CELL DESIGN PROBLEM
CELL FORMATION PROBLEM
CELLULAR MANUFACTURING SYSTEM
MULTI-CRITERION DECISION MAKING
MULTI-OBJECTIVE OPTIMIZATION
GENETIC ALGORIHMS

تشكر و قدرداني

بدينوسيله از تمام بزرگواراني كه مرا در انجام اين رساله ياري كرده اند به ويژه استاد ارجمند جناب آقاي دكتر معطر حسيني كه وظيفه هدايت و راهنمايي رساله را بر عهده داشته اند تشكر و سپاسگزاري مي نمايد. همچنين از كمك هاي فكري آقايان دكتر ذگردي و دكتر قدسي پور كه اساتيد مشاور بنده در مراحل مختلف انجام رساله بوده اند تشكر مي نمايد. از آقاي دكتر نيومن از دانشگاه لافبرو انگلستان كه استاد راهنماي اينجانب در دوره اقامت من در اين دانشگاه بوده اند قدرداني مي گردد. از پدر و مادر ارجمندم كه راهنما، مشوق و حامي من در تمام مراحل تحصيل بوده اند كمال تشكر را مي نمايد. و بالاخره از همسر عزيزم كه صبورانه يار و همراه من در تمام مراحل زندگي مشتركمان به ويژه در دوران انجام رساله حاضر بوده و نيز فرزندان دلبندم كه اشتغال فكري من در اين دوره، فرصت بودن مؤثر در كنار آنها را محدود كرده بود صميمانه سپاسگزاري مي نمايد.

تقدیم به :

پدر، مادر، همسر و فرزندانم

فهرست علائم اختصاري

CPU	**C**entral **P**rocessing **U**nit
CR	**C**rossover **R**ate
CNDF	**C**urrent **N**on-**D**ominated **F**ront
DM	**D**ecision **M**aker
DF	**D**egrading **F**actor
DFV	**D**ummy **F**itness **V**alue
EEVL	**E**dinburgh **E**ngineering **V**irtual **L**ibrary
ES	**E**lite **S**et
ESS	**E**lite **S**et **S**ize
EN	**E**psilon **N**iche
ENS	**E**xpected **N**umber of **S**election
GA	**G**enetic **A**lgorithm
ITP	**I**nitial **T**ransfer **P**robability
IP	**I**nversion **P**robability
MP	**M**ating **P**ool
MS	**M**ating **S**et
MG	**M**aximum number of **G**enerations
MP	**M**easure of **P**erformance
Min_SNDF	**Min**imum **S**uccessive **N**on-**D**ominated **F**ronts
MADM	**M**ulti-**A**ttribute **D**ecision **M**aking
MCDM	**M**ulti-**C**riterion **D**ecision **M**aking
MODM	**M**ulti-**O**bjective **D**ecision **M**aking
MOP	**M**ulti-objective **O**ptimization **P**roblem
MR	**M**utation **R**ate
NN	**N**eural **N**etwork
NC	**N**iche **C**ount
NPGA	**N**iched **P**areto **G**enetic **A**lgorithm
NP	**N**iching **P**arameter
NSGA	**N**on-dominated **S**orting **G**enetic **A**lgorithm
Pop	**Pop**ulation
Pop_Size	**Pop**ulation **Size**
PS	**P**opulation **S**ize
PSelect	**P**robability of **Select**ion
RSSWP	**R**eminder **S**tochastic **S**ampling **W**ithout **R**eplacement
RSSWR_UE	**R**eminder **S**tochastic **S**ampling **W**ithout **R**eplacement **U**sing **E**litism
SFV	**S**hared **F**itness **V**alue
SOP	**S**ingle-objective **O**ptimization **P**roblem
SNDF	**S**uccessive **N**on-**D**ominated **F**ronts
TS	**T**abu **S**earch
TP	**T**ransfer **P**robability
VEGA	**V**ector **E**valuated **G**enetic **A**lgorithm
XGA	**X G**enetic **A**lgorithm

فهرست شکل ها

صفحه

شکل (۱-۱)	جایگاه کاربردي سیستم هاي کارگاهي، جریاني و سلولي	٦
شکل (۲-۲)	طبقه بندي ورودي ها	۹
شکل (۳-۲)	طبقه بندي معیارها	۱۱
شکل (٤-۲)	طبقه بندي روش هاي حل	۱۳
شکل (۵-۲)	طبقه بندي خروجي ها	۱٤
شکل (۱-۳)	یک طرح تفکیک ماتریس قطعه _ ماشین	۱۹
شکل (۲-۳)	ماتریسهاي اولیه و مرتب شده قطعه _ ماشین	۲٦
شکل (۳-۳)	زمانهاي پردازش (t_{ij} ها) و سایر داده هاي مسئله	۲۷
شکل (٤-۳)	چگونگي تغییرات مقادیر نرمالیزه شده توابع هدف	۳۰
شکل (۱-۴)	بیان تصویري بهینگي پارتو در فضاي هدف (چپ) و روابط ممکن بین جواب ها در فضاي هدف (راست)	۳٤
شکل (۲-۴)	سهم کروموزوم ها درچرخ رولت	٤۲
شکل (۳-٤)	طرز کار عملگرهاي انتخاب، ترکیب و جهش در الگوریتم ژنتیک	٤٤
شکل (٤-٤)	شبه کد یک الگوریتم ژنتیک استاندارد	٤۵
شکل (۵-٤)	یک عصب تک ورودي	٤۹
شکل (٦-٤)	یک عصب با ورودي هاي چندگانه	٤۹
شکل (۷-۴)	یک لایه مشتمل بر S عصب	۵۱
شکل (۸-۴)	یک شبکه عصبي سه لایه	۵۲
شکل (۱-۵)	فلوچارت کلي الگوریتم XGA	٦۳
شکل (۲-۵)	فلوچارت به هنگام سازي مجموعه نخبه ها	٦٦
شکل (۳-۵)	فلوچارت طرح انتخاب *RSSWR-UE*	٦۹
شکل (٤-۵)	چگونگي ترکیب دو کروموزوم	۷۰
شکل (۵-۵)	چگونگي کارکرد عملگر معکوس سازي	۷۰
شکل (٦-۵)	فایل ورود داده هاي یک مسئله نمونه	۷۵
شکل (۷-۵)	نمونه خروجي نرم افزار	۷٦
شکل (۱-٦)	چگونگي تغییرات MP_1 در الگوریتم هاي مختلف	۹۲

شکل (٦-٢) چگونگي تغييرات MP_2 در الگوريتم هاي مختلف ٩٢
شکل (٦-٣) چگونگي تغييرات MP_3 در الگوريتم هاي مختلف ٩٢
شکل (٦-٤) چگونگي تغييرات MP١ در مقايسات دودويي الگوريتم ها ٩٥

فهرست جداول

صفحه

جدول (۱-۲) تعدادي از مراجع طراحي سلولهاي توليدي در حالت تک هدفي ۷
جدول (۲-۲) مقايسه مدل ها بر اساس ورودي ها ۱۰
جدول (۳-۲) مقايسه مدل ها بر اساس معيارها ۱۲
جدول (٤-۲) مقايسه مدل ها بر اساس روش حل ۱٤
جدول (۵-۲) مقايسه مدل ها بر اساس خروجي ها ۱۵
جدول (۱-۳) اطلاعات مجموعه هاي مسئله ۲۷
جدول (۲-۳) بردارهاي جواب ممکن و مقادير توابع هدف آنها ۲۹
جدول (۱-۴) مقايسه MADM و MODM ۳۲
جدول (۲-٤) مثالي از يک جامعه و شايستگي اعضاي آن ٤۲
جدول (۱-۵) مشخصات اعضاي کلاس CPopulation ۷۳
جدول (۲-۵) مشخصات اعضاي کلاس CExceptionalPart ۷٤
جدول (۳-۵) مشخصات اعضاي کلاس CCell ۷٤
جدول (٤-۵) پارامترهاي الگوريتم XGA ۸۱
جدول (۱-٦) دامنه پارامترهاي متغير الگوريتم XGA مورد استفاده در توليد مجموعه هاي مرجع ۸۹
جدول (۲-٦) مشخصات مسائل مورد استفاده در ارزيابي الگوريتم ها ۹۰
جدول (۳-٦) نتايج مقايسه با مجموعه هاي نامغلوب مرجع ۹۱
جدول (٤-٦) نتايج مقايسات دودويي الگوريتم ها ۹٤

فهرست مطالب

صفحه

تشكر و قدرداني		الف
چكيده به فارسي		ب
چكيده به انگليسي		ج
فهرست علائم اختصاري		د
فهرست شكل ها		ه
فهرست جداول		ز
فهرست مطالب		ح
فصل ۱	مقدمه	۱
	(۱-۱) تعريف مسئله	۱
	(۱-۲) كاربرد رساله در سطح ملي و بين المللي	۲
	(۱-۳) فرض هاي رساله	۲
	(۱-٤) چارچوب رساله	٤
فصل ۲	مرور ادبيات	۵
	(۲-۱) تعريف مسئله طراحي سلولهاي توليدي	۵
	(۲-۱-۱) معرفي سيستم توليد سلولي	۵
	(۲-۱-۲) مسئله طراحي سلولهاي توليدي	٦
	(۲-۲) مروري بر كارهاي گذشته	۷
	(۲-۳) مروري بر روشهاي چند معياري طراحي سلول	۹
	(۲-۳-۱) مقالات ارائه شده تا سال ۱۹۹۸	۹
	(۲-۳-۱-۱) مقايسه بر اساس ورودي ها	۹
	(۲-۳-۱-۲) مقايسه بر اساس معيارها	۱۱
	(۲-۳-۱-۳) مقايسه بر اساس روش حل	۱۳
	(۲-۳-۱-٤) مقايسه بر اساس خروجي ها	۱٤

(۲-۳-۲) مقالات ارائه شده در سالهاي ۱۹۹۹ و ۲۰۰۰ ۱۵
(٤-۲) جمع بندي ۱٦

فصل ۳ توسعه یک مدل چند هدفي براي طراحي سلولهاي توليدي ۱۸
(۱-۳) تعريف مسئله ۱۸
(۱-۱-۳) فرض ها ۱۹
(۲-۱-۳) مجموعه معيارها ۲۰
(۲-۳) فرمول بندي مسئله ۲۰
(۱-۲-۳) نمادها ۲۱
(۲-۲-۳) توابع هدف ۲۲
(۱-۲-۲-۳) کمينه سازي تعداد حرکات بين سلولي ۲۲
(۲-۲-۲-۳) کمينه سازي هزينه ۲۲
(۳-۲-۲-۳) کمينه سازي عدم بهره گيري از ماشين آلات ۲۳
(٤-۲-۲-۳) کمينه سازي عدم تعادل بار کاري سلولها ۲٤
(۳-۳) محدوديت ها ۲٤
(۱-۳-۳) محدوديت هاي اصلي ۲٤
(۲-۳-۳) محدوديت هاي اختياري ۲۵
(٤-۳) پيچيدگي مدل ۲۵
(۵-۳) مثال ۲٦
(٦-۳) جمع بندي ۲۹

فصل ٤ مروري بر مسئله بهينه سازي چند هدفي و چگونگي استفاده از الگوريتم هاي ژنتيك و شبکه هاي عصبي در حل آن ۳۱
(۱-٤) خانواده مسائل چند معياري ۳۱
(۲-٤) بهينه سازي چند هدفي ۳۳
(۳-٤) جستجو و تصميم گيري ۳۷
(٤-٤) رويکردهاي سنتي ۳۸
(۵-٤) چگونگي استفاده از الگوريتم هاي ژنتيك و شبکه هاي عصبي ۳۹
(۱-۵-٤) الگوريتم هاي ژنتيك ٤۰
(۱-۱-۵-٤) چارچوب يک الگوريتم ژنتيک استاندارد ٤۰

(۲-۱-۵-٤) کاربرد الگوريتم هاي ژنتيک در بهينه سازي ٤٥
(۳-۱-۵-٤) چگونگي اعمال محدوديت ها ٤٦
(٤-۱-۵-٤) بهينه سازي چند هدفي با استفاده از الگوريتم هاي ژنتيک ٤٧
(۲-۵-٤) شبکه هاي عصبي ٤٨
(۱-۲-۵-٤) ساختار يک شبکه عصبي ٤٨
(۲-۲-۵-٤) طراحي شبکه عصبي ٥٢
(۳-۲-۵-٤) کاربرد شبکه هاي عصبي در بهينه سازي ٥٤
(٤-۲-۵-٤) بهينه سازي چند هدفي با کمک شبکه هاي عصبي ٥٦
(٦-٤) جمع بندي ٥٧

فصل ۵ توسعه يک الگوريتم ژنتيک براي حل مدل چند هدفي طراحي سلولهاي توليدي ٥٩
(۱-۵) مقدمه ٥٩
(۲-۵) نمايش کروموزومي ٦٠
(۳-۵) محاسبه مقادير شايستگي انفرادي ٦٠
(٤-۵) الگوريتم XGA ٦١
(۱-٤-۵) نمادگذاري ٦١
(۲-٤-۵) محاسبه مقدار شايستگي تلفيقي ٦٢
(۳-٤-۵) پيشگيري از انتخاب جواب هاي نشدني ٦٤
(٤-٤-۵) به هنگام سازي مجموعه نخبه ٦٥
(٥-٤-۵) انتخاب ٦٥
(٦-٤-۵) مخلوط کردن استخر امتزاج ٦٨
(۷-٤-۵) ترکيب ٦٨
(۸-٤-۵) معکوس سازي ٧٠
(۹-٤-۵) جهش ٧١
(۱۰-٤-۵) کنار گذاشتن جواب هاي نشدني ٧١
(۱۱-٤-۵) معيار توقف ٧١
(٥-۵) نرم افزار کامپيوتري الگوريتم XGA ٧٢
(۱-٥-۵) ساختمان داده ها ٧٢

(۲-۵-۵) ورودي و خروجي هاي نرم افزار ۷۲
(۶-۵) تنظيم پارامترها ۷۶
(۱-۶-۵) مقياس هاي كارآيي ۷۷
(۲-۶-۵) ارزيابي حساسيت الگوريتم نسبت به پارامترها ۷۸
(۷-۵) جمع بندي ۸۲

فصل ۶ ارزيابي الگوريتم XGA ۸۳
(۱-۶) مقدمه ۸۳
(۲-۶) معرفي الگوريتم هاي مبنا ۸۴
(۱-۲-۶) الگوريتم VEGA ۸۵
(۲-۲-۶) الگوريتم NPGA ۸۵
(۳-۲-۶) الگوريتم NSGA ۸۶
(۳-۶) مجموعه پارامترهاي مورد استفاده ۸۷
(۴-۶) نرم افزار و سخت افزار مورد استفاده ۸۸
(۵-۶) روش هاي ارزيابي ۸۸
(۱-۵-۶) روش ۱ : استفاده از يک مجموعه مرجع ۸۸
(۲-۵-۶) روش ۲ : مقايسات دو به دو ۸۹
(۶-۶) مسائل مورد استفاده در ارزيابي ۹۰
(۷-۶) نتايج ارزيابي ۹۰
(۱-۷-۶) نتايج حاصل از روش ۱ ۹۰
(۲-۷-۶) نتايج حاصل از روش ۲ ۹۳
(۸-۶) جمع بندي ۹۶

فصل۷ جمع بندي و نتيجه گيري ۹۷
(۱-۷) تعريف مسئله طراحي سلولهاي توليدي ۹۷
(۲-۷) جمع بندي ۹۸
(۳-۷) زمينه هاي تحقيقاتي آتي ۱۰۳
(۱-۳-۷) توسعه مدل هاي جديد طراحي سلول و حل آنها ۱۰۳
(۲-۳-۷) حل ساير مسائل بهينه سازي چند هدفي ۱۰۴
(۳-۳-۷) توسعه الگوريتم هاي ژنتيک چند هدفي ۱۰۴

فهرست مراجع ۱۰۵

پیوست ۱ نمودارهاي مربوط به تنظيم پارامترهاي الگوريتم XGA ۱۱٤

پیوست ۲ متغيرها و برخي توابع نرم افزار XGA ۱۲۵

پیوست ۳ مسائل مورد استفاده درارزيابي الگوريتم XGA ۱٤٦

پیوست ٤ كپي مقاله : A review of the modern approaches to multi-criteria cell design ۱۵۰

فصل ۱

مقدمه

(۱-۱). تعريف مسئله

سيستم هاي سنتي توليد را از نظر نحوه آرايش تجهيزات و ماهيت وظايف برنامه ريزي و كنترل توليد مي توان به دو دسته سيستم هاي جرياني و سيستم هاي كارگاهي تقسيم بندي نمود. سيستم جرياني براي توليد انبوه تعداد محدودي از محصولات و سيستم كارگاهي براي توليد محصولات متنوع در تعداد كم مناسب مي باشند. سيستم توليد سلولي سيستمي نسبتا" جديد براي پر كردن محدوده خالي مابين فضاي كاري سيستم هاي سنتي كارگاهي و جرياني است، يعني جايي كه تنوع و نرخ تقاضاي محصولات در حد متوسط مي باشند. سيستم سلولي معمولا" از تغيير سازمان سيستم كارگاهي شكل مي گيرد. از مهمترين مزاياي اقتصادي اين تغيير مي توان به موارد زير اشاره نمود:

- كاهش سهم سرمايه ثابت قطعات از طريق بهره گيري بيشتر از ظرفيت هاي ماشيني
- كاهش سرمايه در گردش به دنبال كاهش حجم موجودي مواد اوليه و قطعات نيم ساخته
- كوتاه شدن دوره برگشت سرمايه به خاطر كاسته شدن از طول زمان ساخت

دستيابي به مزاياي سيستم توليد سلولي درگرو سه تغيير اساسي در زمينه هاي: (۱) تغيير نحوه آرايش و سازماندهي ماشين آلات و تشكيل سلولهاي توليدي، (۲) تغيير سيستم سفارش دهي مواد وكنترل موجودي ها و (۳) تغيير سيستم برنامه ريزي توليد مي باشد.

ومرلوف و هير (۱۹۸۷) تغيير اول را جزء مسائل طراحي ساختار سيستم و تغييرات دوم و سوم را در زمره مسائل طراحي ويژگي هاي عملياتي سيستم بر شمرده اند [۸۷]. آنان همچنين انتخاب جامعه قطعات و گروه بندي آنها در قالب خانواده هاي قطعه و انتخاب ماشين ها و فرآيندهاي ساخت و گروه بندي آنها در قالب سلولهاي توليدي را به عنوان دو تصميم اساسي در مرحله طراحي ساختار سيستم توليد سلولي مطرح مي نمايند. شناسايي و تعيين خانواده هاي قطعه و گروه هاي ماشيني در طراحي يک سيستم توليد سلولي به مسئله طراحي سلول هاي توليدي و يا تشكيل سلولهاي توليدي (راجاماني و همكاران ۱۹۹۶) معروف است [٦٤]. هدف اين رساله ارائه مدلي جديد براي مسئله طراحي سلول هاي توليدي با در نظر داشتن اهداف و معيارهاي چندگانه و حل آن مي باشد.

(۲-۱). كاربرد رساله در سطح ملي و بين المللي

سيستم توليد سلولي به عنوان راه حلي موثر براي توليد اقتصادي دامنه وسيعي از قطعات در سطح جهاني مطرح است. با توجه به روند توسعه اقتصادي در ايران و لزوم راهيابي محصولات ايراني به بازارهاي جهاني، استفاده از سيستم هاي توليد اقتصادي نظير سيستم توليد سلولي ضروري است. لذا توسعه روش هايي براي طراحي سيستم توليد سلولي و اجزاي آن با در نظر گرفتن معيارها و ويژگي هاي مهم در فضاي صنعتي ايران اهميت مي يابد. از جمله اين ويژگي ها مي توان به وجود ظرفيت هاي سخت افزاري زياد و كمبود منابع مالي به ويژه در بعد ارزي اشاره نمود. وجود اين ويژگي ها توجه بيشتر به معيارهايي نظير بيشينه سازي بهره گيري از امكانات و كمينه سازي هزينه لازم براي سرمايه گذاري در تجهيزات جديد را ايجاب مي نمايد.

با توجه به اينكه بسياري از كشورهاي در حال توسعه داراي شرايط صنعتي مشابه ايران مي باشند، نتايج اين تحقيق مي تواند براي آنان نيز مفيد باشد. به علاوه نتايج حاصل از توسعه و بهبود روش هاي حل مسائل چند هدفي محدود به محدوده جغرافيايي خاصي نبوده و براي محققين در سراسر جهان قابل استفاده خواهد بود.

(۳-۱). فرض هاي رساله

فرض هاي زير در توسعه مدل طراحي سلول هاي توليدي در نظر گرفته مي شوند:

- سیستم تولید سلولي از آرایش مجدد یک سیستم تولید کارگاهي شکل مي گیرد. قطعات و ماشینهاي موجود در سیستم کارگاهي تشکیل دهنده خانواده هاي قطعه و سلولهاي ماشیني هستند. در نتیجه سیستم سلولي ظرفیت کافي براي تولید همه قطعاتي که قبلا" در سیستم کارگاهي تولید مي شدند را دارا خواهد بود.
- حجم تقاضاي قطعات در دوره برنامه ریزي مشخص مي باشد.
- براي کاهش وابستگي بین سلولها و حرکات بین سلولي از دو سیاست تکرار ماشین هاي گلوگاهي و مقاطعه فرعي قطعات استثنایي مي توان استفاده کرد.
- در صورت اتخاذ سیاست مقاطعه فرعي براي یک قطعه استثنایي، کل تقاضاي آن از طریق مقاطعه فرعي تامین خواهد شد.

همچنین اهداف زیر در انتخاب طرح یا طرح هاي سلولي مد نظر قرار گرفته است:

- کمینه سازي حرکات بین سلولي.
- کمینه سازي مجموع هزینه هاي تکرار ماشین ها و مقاطعه فرعي قطعات استثنایي.
- کمینه سازي نسبت عدم بهره گیري از ماشین ها در سیستم سلولي که معادل است با بیشینه سازي نسبت بهره گیري از ماشین ها.
- کمینه سازي اختلاف بار کاري سلول هاي مختلف.

اهداف فوق متعارض یکدیگر مي باشند. به عنوان مثال براي کمینه سازي تعداد حرکات بین سلولي در مواردي لازم است اقدام به خرید ماشین هاي جدید (تکرار ماشین) نمود. این مسئله با توجه به فرض وجود ظرفیت کافي در سیستم کارگاهي، موجب افزایش نرخ بیکاري ماشین ها (متضاد با هدف بیشینه سازي میزان بهره گیري از ماشین ها)، و افزایش هزینه هاي تبدیل سیستم (معارض هدف کمینه سازي هزینه هاي تبدیل سیستم) خواهد شد. همچنین کاهش اختلاف بار کاري سلولها مستلزم انتقال بخشي از وظایف ماشین ها و یا سلولها به یکدیگر است که این کار موجب افزایش حرکت قطعات بین سلولها شده و لذا در خلاف جهت هدف کمینه سازي حرکات بین سلولي حرکت مي نماید.

با توجه به ماهیت متضاد اهداف مدل پیشنهادي، پیدا کردن جوابي که همزمان تمامي اهداف را بهینه سازد در حالت کلي غیر ممکن است. بنابر این روش حل باید به جستجوي مجموعه جواب هایي باشد

که مجموعه اهداف فوق را به نحو مناسب و رضايت بخشي ارضاء نمايد. بدين منظور ۲ گزينه شبکه هاي عصبي و الگوريتم هاي ژنتيک به عنوان نامزدهاي روش حل بررسي شده و يکي از آنها براي توسعه روش حل انتخاب خواهد شد.

(۴-۱). چارچوب رساله

مطالب رساله حاضر به شرح زير سازماندهي شده است:

ادبيات مسئله طراحي سلولهاي توليدي با تاکيد ويژه بر روشهاي چند معياري در فصل ۲ مرور خواهد شد. در فصل ۳ مسئله چگونگي برخورد با عناصر استثنايي در طراحي يک سيستم توليد سلولي مورد بحث قرار گرفته و يک مدل ۴ هدفي براي پيدا کردن جواب هاي نامغلوب مسئله طراحي سيستم توليد سلولي ارائه مي شود. فصل ۴ به بحث روي مسئله تصميم گيري چند معياري يا MCDM و بررسي نحوه استفاده از الگوريتم هاي ژنتيك و شبکه هاي عصبي در آن اختصاص يافته است. خانواده مسائل چند معياري در دو زيرگروه تصميم گيري بر اساس ويژگي هاي چند گانه يا MADM و تصميم گيري چند هدفي يا MODM دسته بندي شدند. مدل چند هدفي طراحي سلولهاي توليدي که در فصل ۳ ارائه شد با گروه مسائل تصميم گيري چند هدفي همخواني دارد و لذا بحث هاي تکميلي روي اين گروه که به مسائل بهينه سازي چند هدفي يا MOP نيز معروف هستند متمرکز شده است. در ادامه اين فصل تکنيک هاي الگوريتم ژنتيک و شبکه عصبي به عنوان دو نامزد بالقوه در حل مسئله چند هدفي طراحي سلول مورد ارزيابي اجمالي قرار مي گيرند. در فصل ۵ يک الگوريتم ژنتيک بنام XGA براي حل مسئله چند هدفي طراحي سلول ارائه شده و نرم افزار کامپيوتري آن که به زبان Visual C++ تهيه شده توصيف مي گردد. عملکرد الگوريتم XGA در حل مدل ۴ هدفي طراحي سلولهاي توليدي در فصل ٦ مورد ارزيابي قرار خواهد گرفت. در اين ارزيابي، شاخصهاي: کيفيت جواب، پراکندگي جواب ها و زمان محاسبه در تعدادي مسئله طراحي سلول با سه الگوريتم ژنتيك چند هدفي به نامهاي VEGA ، NPGA و NSGA که نوعا" در ادبيات بهينه سازي چند هدفي با کمك الگوريتم هاي ژنتيك به عنوان مبنا انتخاب مي شوند مقايسه شده است. در انتها و در فصل ۷ مطالب ارائه شده در رساله جمع بندي شده و زمينه هاي تحقيقات بعدي پيشنهاد مي گردد.

فصل ۲

مرور ادبيات

(۱-۲). تعريف مسئله طراحي سلولهاي توليدي

(۱-۱-۲). معرفي سيستم توليد سلولي

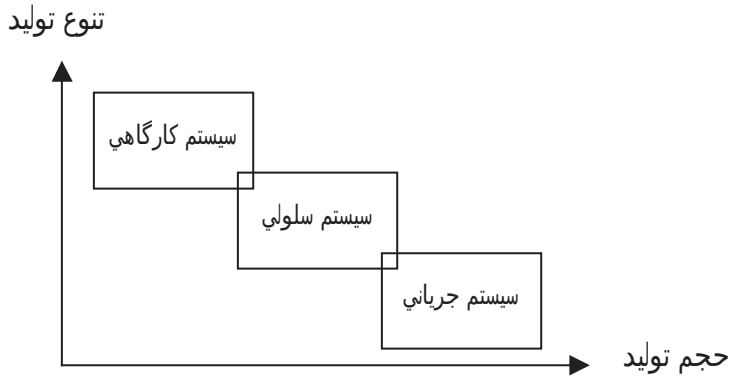
سيستم هاي سنتي توليد را از نظر نحوه آرايش تجهيزات و ماهيت وظايف برنامه ريزي و كنترل توليد به دو دسته سيستم هاي جرياني[۱] و سيستم هاي كارگاهي[۲] تقسيم بندي شده اند (بلک ۱۹۹۱) [۵]. سيستم جرياني براي توليد انبوه تعداد محدودي از محصولات و سيستم كارگاهي براي توليد محصولات متنوع در تعداد كم مناسب مي باشند. سيستم توليد سلولي[۳] سيستمي نسبتا" جديد براي پر كردن محدوده خالي مابين فضاي كاري سيستم هاي سنتي كارگاهي و جرياني است، يعني جايي كه تنوع و نرخ تقاضاي محصولات در حد متوسط مي باشند. شكل (۱-۱) جايگاه سيستم هاي توليدي جرياني، كارگاهي و سلولي را روي نمودار حجم توليد-تنوع توليد نشان مي دهد.

سيستم سلولي معمولا" از تغيير سازمان سيستم كارگاهي شكل مي گيرد (بلک ۱۹۹۱) [۵]. از مهمترين مزاياي اقتصادي اين تغيير مي توان به موارد زير اشاره نمود:

- كاهش سهم سرمايه ثابت قطعات از طريق بهره گيري بيشتر از ظرفيت هاي ماشيني
- كاهش سرمايه در گردش به دنبال كاهش حجم موجودي مواد اوليه و قطعات نيم ساخته

---

[۱] Flow Shop
[۲] Job Shop
[۳] Cellular Manufacturing System



شكل (۱-۱). جايگاه كاربردي سيستم هاي كارگاهي، جرياني و سلولي

- كوتاه شدن دوره برگشت سرمايه به خاطر كاسته شدن از طول زمان ساخت

(۲-۱-۲). مسئله طراحي سلولهاي توليدي

دستيابي به مزاياي سيستم توليد سلولي درگرو سه تغيير اساسي در زمينه هاي زير است (باربيج ۱۹۷۵) [۹]:

- تغيير نحوه آرايش و سازماندهي ماشين آلات و تشكيل سلولهاي توليدي
- تغيير سيستم سفارش دهي مواد وكنترل موجودي ها
- تغيير سيستم برنامه ريزي توليد

ومرلوف و هير (۱۹۸۷) تغيير اول را جزء مسائل طراحي ساختار سيستم[۱] و تغييرات دوم و سوم را در زمره مسائل طراحي ويژگي هاي عملياتي سيستم[۲] بر شمرده اند [۸۷]. آنان همچنين انتخاب جامعه قطعات و گروه بندي آنها در قالب خانواده هاي قطعه[۳] و انتخاب ماشين ها و فرآيندهاي ساخت و گروه بندي آنها در قالب سلولهاي توليدي را به عنوان دو تصميم اساسي در مرحله طراحي ساختار سيستم توليد سلولي مطرح مي نمايند. شناسايي و تعيين خانواده هاي قطعه و گروه هاي ماشيني[٤] در

[۱] System Structure Design
[۲] System Operation Design
[۳] Part Families
[٤] Machine Groups

طراحي يک سيستم توليد سلولي به مسئله طراحي سلول هاي توليدي[۱] و يا تشکيل سلولهاي توليدي[۲] (راجاماني و همکاران ۱۹۹۶) معروف است [٦٤].

(۲-۲). مروري بر کارهاي گذشته

از آغاز دهه ۷۰ ميلادي که مسئله طراحي سلولهاي توليدي مطرح شده، مدل ها و روش هاي حل بسياري براي حل آن ارائه شده است. اهداف و معيارهاي مختلفي در مسئله طراحي سلولهاي توليدي مطرح بوده است. جدول (۱-۲) مهمترين معيارهاي مورد نظر در طراحي سلولهاي توليدي و مقالات برگزيده تا سال ۲۰۰۰ ميلادي که اين معيارها را در مدلسازي مسئله منظور نموده اند را ارائه مي نمايد.

معيار	مراجع منتخب تا سال ۲۰۰۰
کمينه سازي حرکات بين سلولي	کلينسويکس و راجان (۱۹۹۴) [٤٥] ، جوينز و همکاران (۱۹۹۶) [۳۷]، صوفيانوپولو (۱۹۹۷) [۷٤]، چنگ و همکاران (۱۹۹۶) [۱۳]، مون و کيم (۱۹۹۹) [٥٦]، پلاکوئين و پيروال (۲۰۰۰) [٦۲]، کاکس و همکاران (۲۰۰۰) [۱۰]، صوفيانوپولو (۱۹۹۹) [۷٥]
بيشينه سازي شباهت هاي قطعات و/ يا ماشين ها (يا کمينه سازي عدم شباهت ها)	کائو و مون (۱۹۹۱) [۳۸]، کاپارتي و سورش (۱۹۹۱) [۳۹]، چو (۱۹۹۳) [۱٥]، بکتر (۱۹۹۱) [۷]، چن و سريواساتا (۱۹۹۴) [۱۲]، آسکين و همکاران (۱۹۹۱) [۳]، لي و گارسياديار (۱۹۹۳) [٤۹]، کوزياک و همکاران (۱۹۹۳) [٤۷]، کيتائوکا و همکاران (۱۹۹۹) [٤٤]، ون (۲۰۰۰a) [۸۸]، سيف الاسلام و ساركر (۲۰۰۰) [٦۷]
دستيابي به شکل قطري _ بلوکي در ماتريس تلاقي قطعه_ماشين	کاپارتي و سورش (۱۹۹۲) [٤۰]، کاپارتي و همکاران (۱۹۹۳) [٤۱]، سورش و کاپارتي (۱۹۹۴) [۸۰]، موخوپادهياي و همکاران (۱۹۹۴) [٥۸]، چن و ايراني (۱۹۹۳) [۱۱]، دايتا و همکاران (۱۹۹۹) [۱۹]
کمينه سازي عدم تعادل بار کاري سلولها	ونوگوپال و نارندران (b ۱۹۹۲) [۸٤]
کمينه سازي تعداد عناصر استثنايي	سنگ و هيتومي (۱۹۹۲) [۷٦]، اميراحمدي و چوبينه (۱۹۹۶) [۲]، ون (۲۰۰۰b) [۸۹]

جدول (۱-۲). تعدادي از مراجع طراحي سلولهاي توليدي در حالت تک هدفي

همچنين مقالات مروري چندي نيز بر روي مسئله طراحي سلولهاي توليدي ارائه شده است. کينگ و ناکورنچاي (۱۹۸۲) مرور جامعي بر رويکردهاي مختلف مسئله تشکيل سلول تا آن زمان ارائه نموده اند [٤۳]. آنها تمام مدلها را در چهار دسته مبتني بر: ضريب شباهت، نظريه مجموعه ها، ارزيابي کننده[۳]، و ساير مدلهاي تحليلي طبقه بندي کرده اند. موزير و تاوب (۱۹۸٥) ادبيات طراحي سلول تحت چهار

[۱] Manufacturing Cell Design
[۲] Manufacturing Cell Formation
[۳] Evaluative

عنوان: شناسايي قطعات، گروه بندي، زمان بندي در كارگاه مبتني بر تكنولوژي گروهي[۱]، و اجراي تكنولوژي گروهي تفكيک كرده اند [۵۷]. ومرلوف و هير (۱۹۸۶) بيش از ۷۰ مقاله را مرور كرده و آنها را در چهار گروه دسته بندي نموده اند [۸٦]. طرح طبقه بندي آنها تمامي اين روش ها را بر اساس داده هاي اصلي براي گروه بندي، مبتني بر ويژگي هاي قطعات و مبتني بر مسيرهاي توليد قطعات تقسيم كرده اند. گروه اخير به نوبه خود به سه دسته ديگر تقسيم شده است. اين سه دسته عبارتند از: روش هايي كه ابتدا گروه هاي ماشيني و پس از آن خانواده هاي قطعه را شناسايي مي كنند، روش هايي كه نخست به شناسايي خانواده هاي قطعه و سپس به شناسايي گروه هاي ماشيني مي پردازند، و روش هايي كه خانواده هاي قطعه و گروه هاي ماشيني را همزمان شناسايي مي كنند. چو (۱۹۸۹) مرور جامعي روي مبحث توليد سلولي ارائه و ادبيات طراحي سلول را به دو دسته روش هاي طرح مدار[۲] و توليد مدار[۳] تفكيک كرده است [۱٤]. دسته اخير به نوبه خود به زير دسته هاي: آرايه اي[٤] ، سلسله مراتبي، غير سلسله مراتبي، رياضي، نظريه گراف و روش هاي ابتكاري تقسيم شده است. اوفودايل و همكاران (۱۹۹۴) با استفاده از يک چارچوب تاكسونومي[۵] كليه روش هاي شناسايي خانواده هاي قطعه_ماشين را در سه دسته تاكسونومي: روش هاي چشمي، تجزيه و تحليل كد قطعات، و تجزيه و تحليل جريان توليد تقسيم كرده است [٦۰]. روش هاي موجود در دسته اخير تا سال ۱۹۹۱ بر اساس روش حل، متغيرهاي تصميم، اهداف و محدوديت ها با همديگر مقايسه شده اند.

هيچيک از مقالات مروري فوق تعداد معيارهاي منظور شده در مدل طراحي سلول را به عنوان معياري براي دسته بندي مقالات در نظر نگرفته اند. منصوري و همكاران (۲۰۰۰) مدل هاي جديد طراحي سلول را كه بيش از يک معيار را در فرآيند طراحي دخيل كرده اند مرور نموده اند [۵۳]. بخش بعد به مرور اجمالي مدلهاي چند معياري طراحي سلول در دهه ۹۰ بر اساس اين مقاله مي پردازد. براي مطالعات تكميلي در خصوص ويژگي هاي مدلهاي چند هدفي طراحي سلولهاي توليدي و زمينه هاي تحقيقاتي موجود در اين زمينه مي توان به مقاله منصوري و همكاران (۲۰۰۰) كه در پيوست (۴) آمده رجوع نمود.

---

[۱] Group Technology
[۲] Design-Oriented
[۳] Production-Oriented
[٤] Array-Based
[۵] Taxonomic

(۳-۲). مروري بر روشهاي چند معياري طراحي سلول

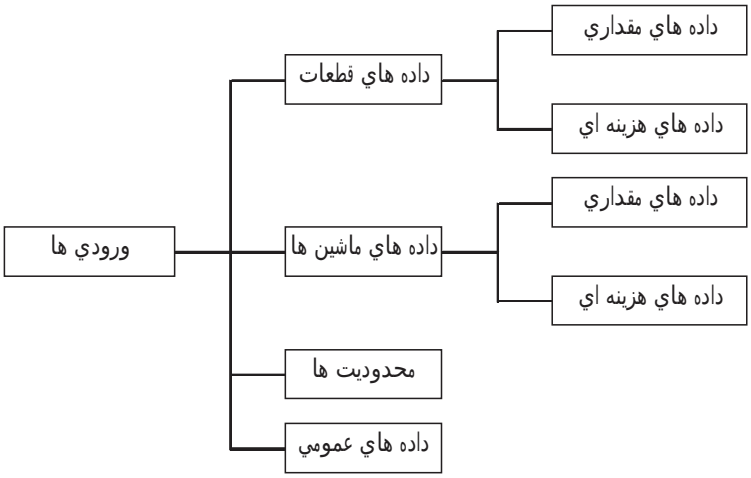
طراحي سلولهاي توليدي با در نظر گرفتن معيارهاي چندگانه يکي از موضوعات تحقيقاتي جذاب در دهه ۹۰ بوده است. اين قسمت به مروري بر روي ويژگي هاي اصلي مدلهاي ارائه شده در اين زمينه مي پردازد. در اين رابطه ابتدا مقالات مرور شده توسط منصوري و همکاران (۲۰۰۰) تا سال ۱۹۹۸ مورد بررسي قرار مي گيرند. در ادامه مقالات ارائه شده در سالهاي ۱۹۹۹ و ۲۰۰۰ مرور خواهند شد. معيار اصلي براي لحاظ کردن يک مقاله در اين مرور، در نظر گرفتن حداقل دو معيار به طور همزمان در روش حل مدل پيشنهادي بوده است.

(۱-۳-۲). مقالات ارائه شده تا سال ۱۹۹۸

در مقاله منصوري و همکاران (۲۰۰۰) مقالات مربوط به طراحي سلول هاي توليدي در حالت چند هدفي تا سال ۱۹۹۸ بر اساس ورودي ها، معيارها، روش هاي حل و خروجي هاي اين مدل ها صورت گرفته است [۵۳]. در اين قسمت نتايج حاصله در اين مقاله مورد بحث قرار مي گيرد.

(۱-۱-۳-۲). مقايسه بر اساس ورودي ها

داده هاي ورودي مدلها ابتدا بر اساس منبع داده ها تقسيم شده اند. آنها به چهار دسته: داده هاي قطعات، داده هاي ماشين ها، محدوديت ها و داده هاي عمومي تفکيک شده اند. داده هاي مربوط به ماشين ها و قطعات بر اساس نوع داده به دو دسته داده هاي مقداري و هزينه اي تقسيم شده اند. شکل (۲-۲) طبقه بندي ورودي ها را نمايش مي دهد.

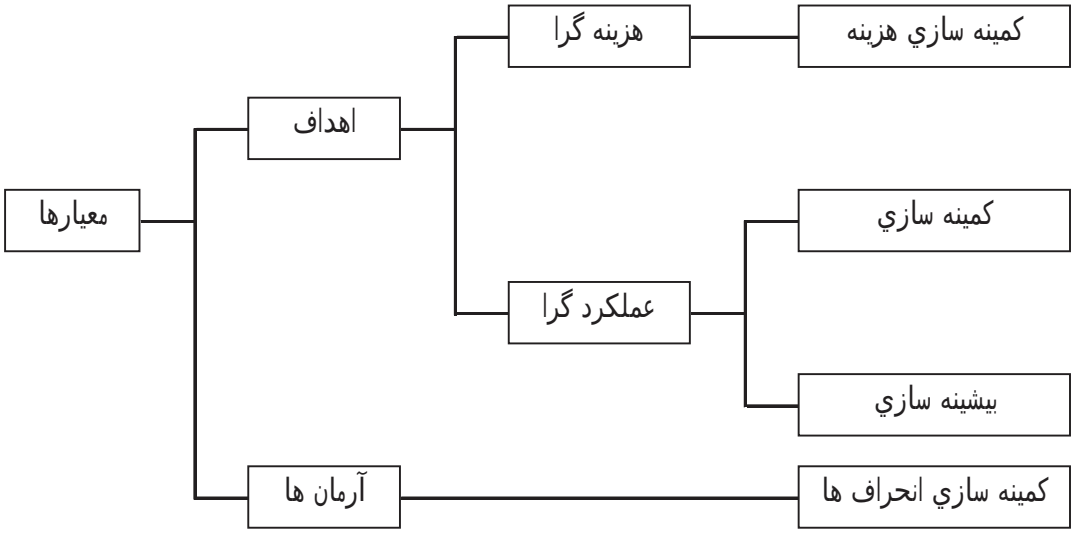


شکل (۲-۲). طبقه بندي ورودي ها

			نويسندگان	ساكاران	وي و گاتير	شافر و راجرز	شافر و همكاران	ونوگوپال و نارندران	داهل و اسميت	لوگدران	مين و شين	ساكاران و كاسيليگام	گوپتا و همكاران	ليانگ و ناون	سورش و همكاران	آكنورك و بالكوز	بكنر	گوپتا و همكاران	هو و مودي	راجماني و همكاران	لي و چن	سو و هسو
			سال	۱۹۹۰	۱۹۹۰	۱۹۹۱	۱۹۹۲	۱۹۹۲	۱۹۹۳	۱۹۹۳	۱۹۹۳	۱۹۹۳	۱۹۹۵	۱۹۹۵	۱۹۹۵	۱۹۹۶	۱۹۹۶	۱۹۹۶	۱۹۹۶	۱۹۹۶	۱۹۹۷	۱۹۹۸
			شماره مرجع	٦٨	٨٥	٧١	٧٢	٨٣	١٨	٥٢	٥٥	٦٩	٢٥	٥٠	٨١	١	٨	٢٦	٢٩	٦٤	٤٨	٧٩
ورودي ها	داده هاي قطعات	مقداري	تقاضا	*	*	*	*	*	*			*			*	*	*	*	*	*	*	
			ماشين هاي مورد نياز	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
			ابزارهاي مورد نياز	*																		
			مسير ثابت توليد			*			*	*		*	*		*	*						*
			مسيرهاي توليد جايگزين																*	*		
			ماشين هاي جايگزين براي عمليات																*	*		
			زمان هاي پردازش	*	*	*	*	*	*	*	*	*	*		*	*	*	*	*	*	*	*
			زمان هاي آماده سازي						*						*						*	*
			اندازه دسته			*			*						*						*	
			ظرفيت پالت																		*	
			كد قطعه													*						
		هزينه اي	هزينه مستقيم توليد	*								*							*	*		
			هزينه استفاده از ابزار	*																		
			دستمزد اپراتورها								*											
			هزينه مقاطعه فرعي				*															
			هزينه حمل و نقل بين سلولي				*					*					*		*	*		*
			هزينه حمل و نقل درون سلولي	*								*										*
			هزينه آماده سازي									*										
			هزينه فضاي كاري									*										
	داده هاي ماشين ها	مقداري	تعداد موجود از هر ماشين			*			*	*		*	*	*	*		*	*	*			
			ظرفيت توليد	*	*	*	*	*	*	*		*	*		*	*	*	*	*	*	*	*
			ابزارهاي موجود	*																		
			فضاي كاري مورد نياز									*										
		هزينه	هزينه خريد	*		*	*									*	*		*	*		
			هزينه عمليات	*												*			*			
	محدوديت ها		تعداد ثابت سلول ها	*	*	*		*	*	*						*	*	*		*		
			دامنه تعداد سلول ها										*						*			
			حداكثر بهره گيري													*						
			حداقل بهره گيري										*		*	*						
			حداكثر تعداد سلول ها	*	*	*			*	*	*			*	*	*	*	*	*	*	*	*
			حداقل تعداد سلول ها			*			*						*		*	*				
			حداكثر خروجي سلول																*			
			محدوديت هاي اپراتور								*											
			بودجه در دسترس			*																
			فضاي در دسترس									*										
	عمومي		مهارت هاي اپراتورها								*											
			آرايش ماشين ها در سلول ها																			*
			آرايش سلول ها در كارخانه															*				*

جدول (۲-۲). مقايسه مدل ها بر اساس ورودي ها

داده هاي ورودي مدل هاي انتخابي در جدول (۲-۲) با هم مقايسه شده اند. سطرهاي اين جدول نماينده ورودي هاي مختلف مدلهاي چند هدفي طراحي سلول مي باشد. رايج ترين داده ورودي اين مدلها: ماشينهاي مورد نياز قطعات، زمانهاي پردازش، ظرفيت توليدي ماشين ها، حداكثر اندازه سلول، و تقاضاي قطعات بوده است. مي توان گفت يک مدل پايه طراحي سلول بايد حداقل چنين مجموعه اي



شكل (۳-۲). طبقه بندي معيارها

از ورودي ها را پوشش دهد. ساير ورودي هاي نسبتا" رايج شامل تعداد ماشين هاي در دسترس، تعداد از پيش تعيين شده سلول ها، جريان توليد ثابت قطعات، هزينه خريد ماشين ها، هزينه حمل و نقل بين سلولي، و حداقل اندازه سلول مي باشد. ورودي هايي نظير زمان هاي آماده سازي، اندازه دسته، هزينه مستقيم توليد، هزينه حمل و نقل درون سلولي، و حد اقل بهره گيري مورد قبول از جمله ورودي هاي كم اهميت تر به شمار مي روند. همچنين ورودي هايي نيز تنها در يک مدل استفاده شده اند كه از آن جمله مي توان به ابزارهاي مورد نياز براي قطعات، هزينه هاي مقاطعه فرعي، فضاي مورد نياز ماشين ها و محدوديت هاي اپراتورها را اشاره نمود.

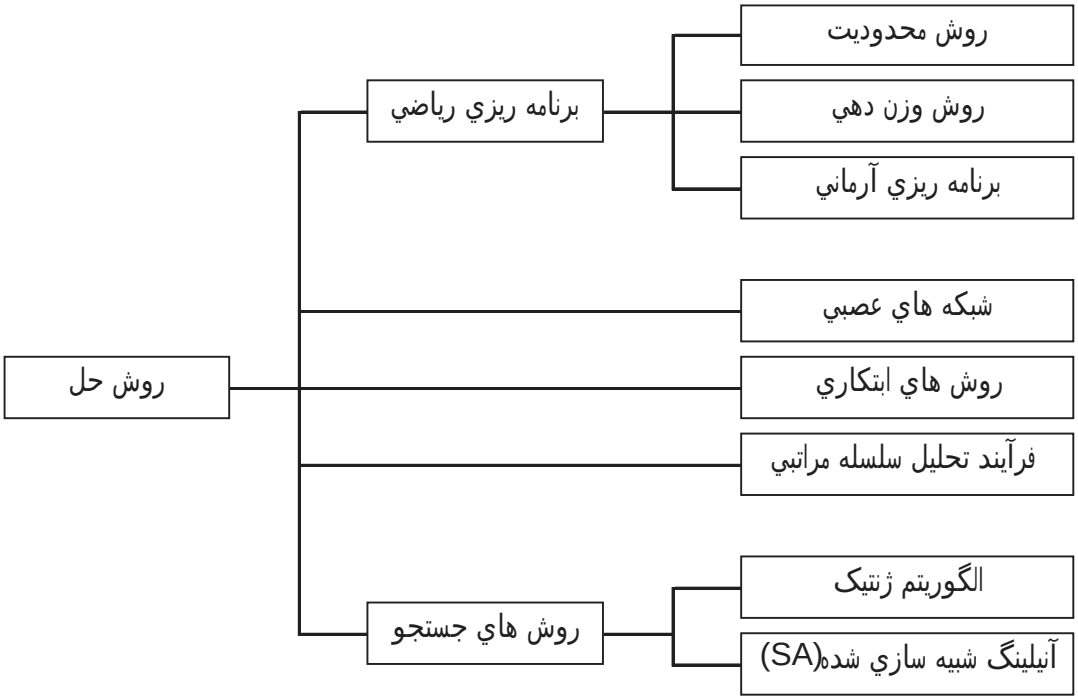
(۲-۱-۳-۲). مقايسه بر اساس معيارها

معيارهاي مورد استفاده در طراحي سلول ابتدا به دو دسته اهداف و آرمانها تقسيم شده اند. اهداف نيز بر اساس گرايش به دو دسته هزينه گرا و عملكرد گرا تفكيك شده اند. اهداف هزينه گرا از نوع كمينه سازي و اهداف عملكرد گرا از هر دو نوع كمينه سازي و بيشينه سازي مي باشند. مدلهاي برنامه ريزي آرماني نيز به كمينه سازي ميزان انحراف از آرمانهاي از پيش تعيين شده مي پردازند. ساختار اين دسته بندي در شكل (۳-۲) نشان داده شده است. جدول (۳-۲) مقايسه اي از مدلها بر اساس اين معيارها ارائه مي دهد. پراكندگي زيادي در معيارهاي مورد استفاده توسط محققين مشاهده مي شود. كمينه سازي هزينه تكرار ماشين، كمينه سازي هزينه حمل و نقل بين سلولي، كمينه سازي حركات

نويسندگان				سانكاران	وي و گاينر	شافر و راجرز	شافر و همكاران	ووگوبال و نارندران	داهل و اسميت	لوگندران	مين و شين	ساكاران و كاسبيلگام	گوپتا و همكاران	ليانگ و ناون	سورش و همكاران	آكيورك و بالكوز	بكنر	گوپتا و همكاران	هو و مودي	راجاماني و همكاران	اي و چن	سو و هسو
سال				۱۹۹۰	۱۹۹۰	۱۹۹۱	۱۹۹۲	۱۹۹۲	۱۹۹۳	۱۹۹۳	۱۹۹۳	۱۹۹۳	۱۹۹۵	۱۹۹۵	۱۹۹۵	۱۹۹۶	۱۹۹۶	۱۹۹۶	۱۹۹۶	۱۹۹۶	۱۹۹۷	۱۹۹۸
شماره مرجع				٦٨	٨٥	٧١	٧٢	٨٣	١٨	٥٢	٥٥	٦٩	٢٥	٥٠	٨١	١	٨	٢٦	٢٩	٦٤	٤٨	٧٩
معيارها	اهداف	هزينه گرا	كمينه سازي هزينه	تكرار ماشين			*									*	*		*	*		*
				هزينه عمليات								*							*	*		
				مقاطعه فرعي قطعه	*		*															
				حمل و نقل بين سلولي	*							*					*		*	*		*
				حمل و نقل درون سلولي			*					*										*
				استفاده از فضا								*										
		عملكرد گرا	كمينه سازي مقدار	ماشين هاي تكرار شده											*							
				حركات بين سلولي				*	*	*			*		*			*				
				حركات درون سلولي						*			*					*			*	
				عدم تعادل بار كاري در سلول ها	*			*								*		*				*
				عدم تعادل بار كاري در كارخانه	*											*					*	*
				عدم تشابه قطعات												*						
				پرش ها												*						
			بيشينه سازي مقدار	انعطاف پذيري					*					*	*							
				كارايي										*								
				بهره گيري	*					*												
				استقلال سلول ها											*							
	آرمان ها	كمينه سازي انحراف از		تشابه قطعات	*						*											
				زمان پردازش در دسترس ماشين ها	*						*											
				دستمزد اپراتورهاي سلول ها							*											
				تطابق مهارت هاي اپراتورها با ماشين ها							*											
				حد اقل حركات قطعات	*																	
				حد اكثر حركات قطعات	*																	
				هزينه سرمايه گذاري در ماشين آلات	*																	
				هزينه عملياتي	*																	
				زمان اماده سازي		*																
				بهره گيري		*																
				بودجه در دسترس		*																
				حركات بين سلولي		*																

جدول (۳-۲). مقايسه مدل ها بر اساس معيارها

بين سلولي، و كمينه سازي عدم تعادل بار كاري سلولها رايج ترين معيارها بوده اند. در يك درجه اهميت پايين تر، معيارهايي چون كمينه سازي هزينه حركات درون سلولي و بيشينه سازي انعطاف پذيري قرار دارند. همچنين مجموعه اي از معيارهاي منحصر به فرد به ويژه در قالب آرمانها وجود دارند. به عنوان مثال مي توان به كمينه سازي انحراف از سطح مورد قبول تطابق ميان مهارت هاي اپراتورها و قابليت ماشين ها، كمينه سازي انحراف از يك سطح از پيش تعيين شده براي سرمايه گذاري در ماشين آلات، و كمينه سازي تعداد پرش ها نام برد. گذشته از پراكندگي معيارها ، مدلهاي با مجموعه يكساني از معيارها نيز بسيار نادربوده است. درواقع تنها هو و مودي (١٩٩٦) [٢٩] وراجاماني و همكاران (١٩٩٦) [٦٤] مجموعه مشابهي از معيارها را در مدلهاي خود منظور نموده اند. لذا مي توان نتيجه گيري كرد كه بيشتر مدلهاي طراحي سلول چند هدفي منحصر به فرد بوده و قابل مقايسه با يكديگر نمي باشند.



شكل (۴-۲). طبقه بندي روش هاي حل

(۳-۱-۳-۲). مقايسه بر اساس روش حل

رويكرد هاي حل بر اساس تكنيك مورد استفاده در گروههايي كه شكل (٤-۲) نشان مي دهد طبقه بندي شده اند. طبقه بندي شاخه برنامه ريزي رياضي به: روش محدوديت، روش وزن دهي، و برنامه ريزي آرماني با الهام از تقسيم بندي هوانگ و يون (۱۹۸۱) [٣٦] بوده است. روش حل مورد استفاده در حل مدلهاي چند هدفي طراحي سلول در جدول (٤-۲) با هم مقايسه شده است. برنامه ريزي رياضي رايج ترين تكنيك مورد استفاده بوده است كه در اين ميان نقش روش وزن دهي و برنامه ريزي آرماني بيشتر بوده است. كاربردهاي برنامه ريزي آرماني مربوط به پيش از سال ۱۹۹۵ بوده و چنين به نظر مي رسد كه اين تكنيك جذابيت خود را در نيمه دوم دهه ۹۰ از دست داده است. استفاده از روش محدوديت تنها در يك مورد گزارش شده است. به دليل ماهيت NP-Complete مسئله طراحي سلول، روش هاي ابتكاري[۱] به طور گسترده در حل مسائل واقعي به كار گرفته شده اند. الگوريتم هاي ژنتيك[۲] و آنيلينگ شبيه سازي شده[۳] تنها روش هاي جستجوي گزارش شده در

[۱] Heuristics
[۲] Genetic Algorithms
[۳] Simulated Annealing

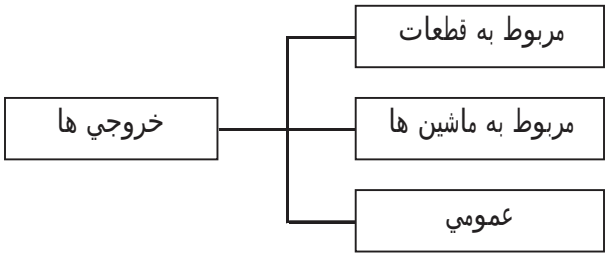
نويسندگان			سانكاران	وي و گاينر	شافر و راجرز	شافر و همكاران	ونوگوپال و ناريدران	داهل و اسميت	اوگدران	مين و شين	ساكاران و كاسپلينگام	گوپتا و همكاران	ليانگ و تاون	سورش و همكاران	آكتورك و بالكور	بكتر	گوپتا و همكاران	هو و مودي	راجاماني و همكاران	لي و چن	سو و هسو
سال			۱۹۹۰	۱۹۹۰	۱۹۹۱	۱۹۹۲	۱۹۹۲	۱۹۹۳	۱۹۹۳	۱۹۹۳	۱۹۹۳	۱۹۹۵	۱۹۹۵	۱۹۹۵	۱۹۹٦	۱۹۹٦	۱۹۹٦	۱۹۹٦	۱۹۹٦	۱۹۹۷	۱۹۹۸
شماره مرجع			٦۸	۸۵	۷۱	۷۲	۸۳	۱۸	۵۲	۵۵	٦۹	۲۵	۵۰	۸۱	۱	۸	۲٦	۲۹	٦٤	٤۸	۷۹
روش حل	برنامه ريزي رياضي	روش محدوديت						*													
		روش وزن دهي		*		*			*		*		*			*		*	*		
		برنامه ريزي آرماني	*		*					*				*							
	شبكه هاي عصبي													*							
	روش هاي ابتكاري			*	*			*		*	*		*		*			*		*	
	فرآيند تحليل سلسله مراتبي(AHP)														*						
	روشهاي جستجو	الگوريتم ژنتيک					*					*					*				
		آنيلينگ شبيه سازي شده (SA)														*					*

جدول (۴-۲). مقايسه مدل ها بر اساس روش حل

حل مدلها بوده اند. شبكه هاي عصبي[1] و فرآيند تحليل سلسله مراتبي[2] نيز به عنوان بخشي از روش حل در كنار ساير تكنيك ها مورد استفاده قرار گرفته اند.

(٤-۱-۳-۲). مقايسه بر اساس خروجي ها

بيشتر مدلهاي طراحي سلول مورد بحث، اطلاعاتي را افزون بر كميت هاي مورد محاسبه در قالب معيارها توليد مي نمايند. اين خروجي هاي اضافي بر اساس ارتباطشان با موضوع گروه بندي تقسيم بندي شده اند. اين تقسيم بندي در شكل (۵-۲) نمايش داده شده است.



شكل (۵-۲). طبقه بندي خروجي ها

[1] Neural Networks
[2] Analytic Hierarchy Process

نويسندگان			سانكاران	وي و گاينر	شافر و راجرز	شافر و همكاران	ونوگوپال و نارندران	داهل و اسميت	لوگدران	مين و نسن	سانكاران و كاسيليگام	گوپتا و همكاران	لياگ و تايون	سورش و همكاران	آكنورك و بالكوز	بكنر	گوپتا و همكاران	هو و مودي	راجاماني و همكاران	لي و چن	سو و هسو
سال			۱۹۹۰	۱۹۹۰	۱۹۹۱	۱۹۹۲	۱۹۹۲	۱۹۹۳	۱۹۹۳	۱۹۹۳	۱۹۹۳	۱۹۹۵	۱۹۹۵	۱۹۹۵	۱۹۹۶	۱۹۹۶	۱۹۹۶	۱۹۹۶	۱۹۹۶	۱۹۹۷	۱۹۹۸
شماره مرجع			٦٨	٨٥	٧١	٧٢	٨٣	١٨	٥٢	٥٥	٦٩	٢٥	٥٠	٨١	١	٨	٢٦	٢٩	٦٤	٤٨	٧٩
خروجي ها (اثرون بر مقادير محاسبه شده به عنوان معيارها)	مربوط به قطعات	خانواده هاي قطعه	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
		قطعات براي مقاطعه فرعي		*		*							*								
		فرايندهاي توليد مختلف																	*		
		حركات بين سلولي				*					*										*
		حركات درون سلولي									*										*
	مربوط به ماشين ها	گروه هاي ماشيني	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
		ماشينهاي تكرار شده				*									*	*		*	*	*	*
		آرايش ماشين ها درون سلول ها													*						*
		بهره گيري										*					*				*
	عمومي	توالي قطعات			*																
		گروه هاي انساني								*											
		آرايش ماشين ها در كارخانه																			*

جدول (۵-۲). مقايسه مدل ها بر اساس خروجي ها

مقايسه مدلها بر اساس خروجي ها در شكل (۵-۲) ارائه شده است. واضح است كه خانواده هاي قطعه و گروههاي ماشيني از جمله خروجي هاي اصلي هر مدل طراحي سلول مي باشند. ماشينهاي تكرار شده، حركات بين سلولي و بهره گيري از ماشينها در زمره خروجي هاي رايج تر بوده اند. ماشينهاي تكرار شده تا نيمه اول دهه ۹۰ يك خروجي اصلي به حساب نمي آمد لكن از اين زمان به بعد تبديل به يك خروجي اصلي شده است. همچنين تعدادي از خروجي هاي منحصر به فرد نظير فرآيندهاي توليد چندگانه، توالي قطعات و گروههاي انساني نيز در موارد خاص مطرح شده اند كه نمي توان آنها را به عنوان يك خروجي هاي اصلي در نظر گرفت.

(۲-۳-۲). مقالات ارائه شده در سالهاي ۱۹۹۹ و ۲۰۰۰

پس از مقاله مروري منصوري و همكاران (۲۰۰۰) [۵۳] كارهاي ديگري در زمينه طراحي سلولهاي توليدي در حالت چند هدفي ارائه شده است. اين بخش به مرور توابع هدف و روشهاي مورد استفاده در حل اين مدلها مي پردازد.

نائير و نارندران (۱۹۹۹) يك مدل دو هدفي براي تشكيل سلولهاي توليدي با اهداف كمينه سازي اختلاف بار كاري در داخل سلولها و كمينه سازي حركات درون سلولي ارائه كردند [۵۹]. آنها با

تعريف يك ضريب شباهت جديد ميان جفتهاي ماشيني، الگوريتمي مبتني بر روش دسته بندي غير سلسله مراتبي[۱] براي حل اين مدل ارائه نموده اند.

شانكر و ورات (۱۹۹۹) يك مدل سه هدفي شامل: كمينه سازي مجموع هزينه هاي حمل و نقل بين سلولي، خريد ماشينهاي جديد و مقاطعه فرعي قطعات، كمينه سازي ظرفيت استفاده نشده از ماشينهاي گلوگاهي و كمينه سازي مجموع حمل و نقل بين سلولي ارائه نموده و در حل آن از برنامه ريزي فازي كمك گرفته اند [۷۳].

ژائو و وو (۲۰۰۰) يك مدل سه هدفي با اهداف كمينه سازي هزينه حركات درون سلولي و بين سلولي، كمينه سازي مجموع اختلاف بار كاري ماشين آلات درون سلولها و كمينه سازي عناصر استثنايي ارائه نموده و براي حل آن يك الگوريتم ژنتيك توسعه دادند [۹۱]. مدل آنها قادر به پذيرش مسيرهاي چندگانه براي توليد قطعات مي باشد.

بايكاس اوغلو و گيندي (۲۰۰۰) براي تبديل يك سيستم كارگاهي به سيستم سلولي، يك مدل برنامه ريزي آرماني شامل اهداف: كمينه سازي عدم شباهت قطعات در سلولها، كمينه سازي مجموع اختلاف بار كاري سلولها، كمينه سازي ظرفيت هاي اضافي مورد نياز در تبديل سيستم كارگاهي به سلولي و بيشينه سازي انعطاف پذيري سلولها ارائه كردند [٤]. آنها براي حل اين مدل از روش جستجوي ممنوع[۲] چند هدفي استفاده نموده اند.

(۲-۴). جمع بندي

در اين فصل ادبيات مسئله طراحي سلولهاي توليدي با تاكيد ويژه بر روشهاي چند معياري مرور شد. به اين منظور ابتدا سيستم توليد سلولي معرفي شده و جايگاه آن در مقايسه با سيستم هاي سنتي كارگاهي و جرياني بر روي نمودار حجم توليد _ تنوع توليد مشخص گرديد. سپس مسئله عمومي طراحي سلولهاي توليدي معرفي شده و مروري بر كارهاي گذشته در حل اين مسئله در حالت تک هدفي ارائه شده و مقالات منتخبي در اين ارتباط معرفي شدند. در ادامه مدلهاي چند معياري طراحي سلولهاي توليدي كه زمينه اصلي اين رساله است با استفاده از يک طرح ابتكاري مقايسه شدند. در اين

[۱] Non-Hierarchical Clustering
[۲] Tabu Search

طرح مقايسه مابين مدلهاي طراحي سلول بر اساس: ورودي ها، معيارها، روش حل، و خروجي هاي مدلها انجام مي شود. نتيجه اين مقايسه و بررسي مقالات منتشره تا سال ۲۰۰۰ نشان مي دهد كه تاكنون مدلي براي طراحي سلولهاي توليدي با اهداف ۴ گانه مورد نظر در اين رساله ارائه نشده است.

فصل ۳

توسعه یک مدل چند هدفي براي طراحي سلولهاي توليدي

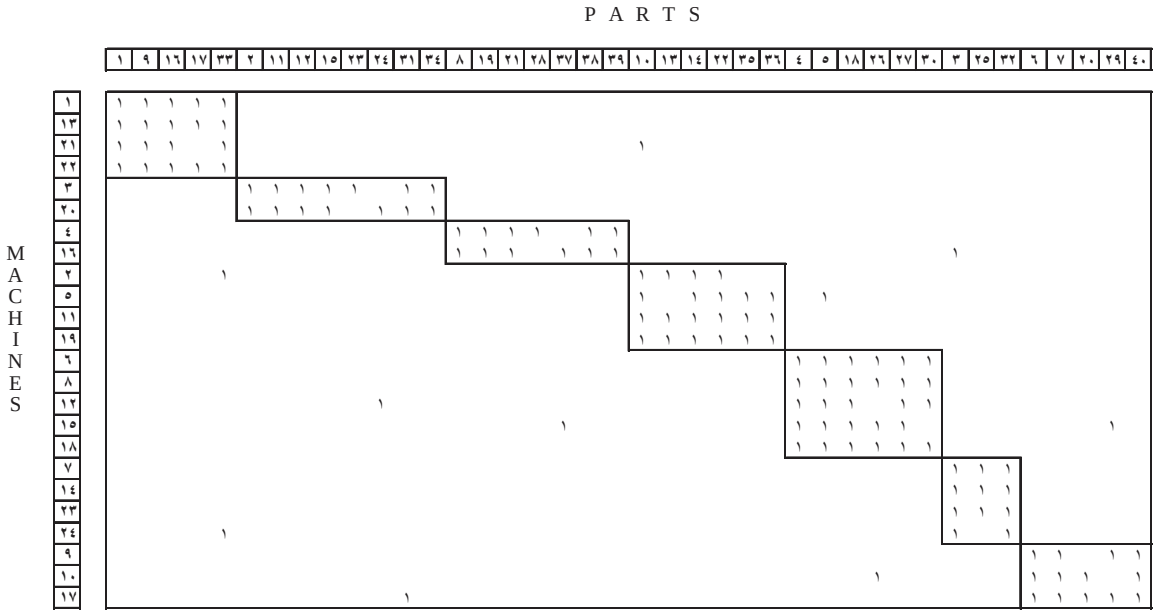
(۱-۳). تعريف مسئله

يكي از مشكلات همراه سيستم هاي توليد سلولي وجود عناصر استثنايي[۱] است. یک ماشين استثنايي ماشيني است كه در سلول (هاي) ديگري به غير از سلولي كه به آن تخصيص يافته مورد نياز است. به چنين ماشيني ماشين گلوگاهي[۲] نيز مي گويند. یک قطعه استثنايي قطعه اي است كه فرآيند ساخت آن در یک سلول تكميل نمي شود و به ماشين يا ماشين هايي در سلولهاي ديگر نيازمند است. شكل (۱-۳) یک طرح تفكيک ماتريس تلاقي قطعه – ماشين مشتمل بر ۷ سلول را نشان مي دهد. در اين طرح تعدادي عنصر استثنايي وجود دارند كه مانع تشكيل سلولهاي كاملا" مستقل مي باشند. به عنوان مثال قطعه ۳۳ كه به خانواده قطعه اول (بالا – چپ) تعلق دارد، به ماشين ۲ نيز احتياج دارد كه اين ماشين در سلول اول موجود نيست وبه سلول چهارم تخصيص يافته است. اين مسئله موجب وايستگي بين سلولهاي اول و چهارم شده و مشكلاتي را در عمليات روزمره اين سلولها ايجاد مي نمايد.

براي رفع وابستگي ناشي از عناصر استثنايي ۲ راه حل پيشنهاد شده است: تكرار ماشين هاي گلوگاهي[۳]

---

[۱] Exceptional Elements
[۲] Bottleneck Machine
[۳] Machine Duplication



شکل (۱-۳). یک طرح تفکیک ماتریس قطعه – ماشین
(برگرفته از مقاله کازروني و همکاران (۱۹۹۷) [٤٢])

و مقاطعه فرعي قطعات استثنايي[1] (شافر و همکاران ۱۹۹۲ [۷۲]). هر یک از این راه حلها ضمن آنکه وابستگي سلولها به یکدیگر را از طریق کم کردن حرکات بین سلولي کاهش مي دهند، تاثیرات نامطلوبي روي سایر پارامترهاي سیستم نظیر هزینه استقرار سیستم و نسبت بهره گیري از ماشین ها مي گذارند. این مسئله همراه با تعداد زیاد ترکیبات ممکن براي تکرار ماشین هاي گلوگاهي و مقاطعه فرعي قطعات استثنايي، استفاده از مدلهاي علمي تصمیم گیري چند معیاري را ضروري مي نماید. در این فصل یک مدل ۴ هدفي براي حل مسئله چگونگي برخورد با عناصر استثنايي ارائه شده است.

(۱-۱-۳). فرض ها

فرض هاي زیر در توسعه مدل در نظر گرفته شده است:

- سیستم توليد سلولي از آرایش مجدد یک سیستم توليد کارگاهي[2] شکل مي گیرد. قطعات و ماشینهاي موجود در سیستم کارگاهي تشکیل دهنده خانواده هاي قطعه و سلولهاي ماشیني

[1] Part Subcontracting
[2] Job Shop

هستند. در نتيجه سيستم سلولي ظرفيت كافي براي توليد همه قطعاتي كه قبلا" در سيستم كارگاهي توليد مي شدند را دارا خواهد بود.

- يک آرايش اوليه از ماتريس تلاقي قطعه – ماشين حاصل از يک روش ماتريسي در اختيار است كه در آن تشكيل تعدادي سلول ماشيني و خانواده قطعه پيشنهاد شده است. در صورتي كه از يک ماشين گلوگاهي بيش از يک دستگاه موجود باشد، تخصيص اين ماشين ها ميان سلولها به گونه اي صورت گرفته كه بخش صحيح تعداد مورد نياز هر سلول تامين شده است. در نتيجه كمبود ظرفيت هر ماشين گلوگاهي در يک سلول كوچكتر يا مساوي ۱ مي باشد.
- براي كاهش وابستگي بين سلولها و حركات بين سلولي از دو سياست تكرار ماشين هاي گلوگاهي و مقاطعه فرعي قطعات استثنايي مي توان استفاده كرد.
- در صورت اتخاذ سياست مقاطعه فرعي براي يک قطعه استثنايي، كل تقاضاي آن از طريق مقاطعه فرعي تامين خواهد شد.

(۲-۱-۳). مجموعه معيارها

معيارهاي زير براي انتخاب طرح يا طرح هاي مناسب سلولي مد نظر قرار گرفته است:

- كمينه سازي حركات بين سلولي.
- كمينه سازي مجموع هزينه هاي تكرار ماشين ها و مقاطعه فرعي قطعات استثنايي.
- كمينه سازي نسبت عدم بهره گيري از ماشين ها در سيستم سلولي كه معادل است با بيشينه سازي نسبت بهره گيري از ماشين ها.
- كمينه سازي اختلاف بار كاري سلول هاي مختلف.

بر اساس مطالعات انجام شده روي مقالات ارائه شده در زمينه طراحي سلول هاي توليدي در حالت چند معياري كه نتايج آن در فصل ۲ ارائه شد، تاكنون مدلي با مجموعه معيارها و فرض هاي فوق گزارش نشده است.

(۲-۳). فرمول بندي مسئله

در اين بخش نمادها و توابع هدف همچنين محدوديت هاي مدل چند هدفي طراحي سلول بر اساس يک جواب اوليه حاصل از يک روش ماتريسي ارائه مي شوند.

(۱-۲-۳). نمادها[۱]

- مجموعه اندیس ها

i : اندیس انواع ماشین، $i = ۱,...,m$

j : اندیس انواع قطعه، $j = ۱,...,p$

k : اندیس سلولها، $k = ۱,...,c$

- مجموعه متغیرهاي تصمیم

X_j : متغیر تصمیم مربوط به مقاطعه فرعي قطعه j ($X_j = ۱$ اگر قطعه j از طریق مقاطعه فرعي تامین شود و $X_j = ۰$ در غیر اینصورت)

Y_{ik} : متغیر تصمیم ناظر بر تکرار ماشین i در سلول k ($Y_{ik} = ۱$ اگر ماشین i در سلول k تکرار شود و $Y_{ik} = ۰$ در غیر اینصورت)

- مجموعه پارامترها
  - داده هاي مربوط به قطعات

D_j : تقاضاي سالیانه براي قطعه j

S_j : هزینه افزایشي مقاطعه فرعي یک واحد از قطعه j

t_{ij} : زمان پردازش یک واحد قطعه j روي ماشین i

PM_{ji} : تعداد حرکات بین سلولي مورد نیاز قطعه j به علت نبودن ماشین i در سلول توليدي این قطعه.

  - داده هاي مربوط به ماشین ها

M_i : هزینه سالیانه بکارگیري یک ماشین اضافي i

CM_i : ظرفیت ماشینکاري هر واحد ماشین i در سال (بر حسب دقیقه)

  - داده هاي مربوط به سلول ها

HF_k : مجموعه قطعات تخصیص یافته به سلول k

MC_k : مجموعه ماشین هاي تخصیص یافته به سلول k

---

[۱] Notation

GF_k : مجموعه قطعاتي که به سلول هاي ديگري به جز k تخصيص يافته اند لکن به برخي از ماشين هاي سلول k نياز دارند

BM_k : مجموعه ماشين هاي گلوگاهي در سلول k

EP_k : مجموعه قطعات استثنايي در سلول k

EM_j : مجموعه ماشين هاي گلوگاهي مورد نياز قطعه استثنايي j

CS_k : تعداد ماشين هاي تخصيص يافته به سلول k (اندازه سلول)

MCS : حداکثر تعداد ماشين هاي قابل تخصيص به يک سلول

(۳-۲-۲). توابع هدف

در اين قسمت اهداف ۴ گانه مورد نظر در ساخت مدل مورد بحث قرار گرفته و فرمول بندي شده اند.

(۳-۲-۲-۱). کمينه سازي تعداد حرکات بين سلولي

يکي از بزرگترين مشکلات يک طرح سلولي حاوي عناصر استثنايي، حرکات بين سلولي است. اين مسئله موجب پيچيدگي جريان قطعات در سطح کارخانه شده و وظيفه برنامه ريزي و کنترل توليد و موجودي ها را مشکل مي سازد. اين امر به نوبه خود موجب عدم بهره گيري کامل از مزاياي طرح سلولي مي شود. به عبارت ديگر يکي از مزاياي عمده طرح سلولي نسبت به طرح کارگاهي[۱] که همانا ساده بودن جريان مواد در سطح کارگاه است در شرايطي که حرکات بين سلولي قابل ملاحظه اي وجود داشته باشد، به طور کامل محقق نمي شود. لذا کمينه سازي تعداد حرکات بين سلولي از طريق تابع هدف ذيل مد نظر قرار گرفته است:

$$Min \quad f_1 = \sum_{k=1}^{c} \sum_{j \in EP_k} (1 - X_j) . \sum_{i \in EM_k} PM_{ji} . (1 - Y_{ik}) \qquad (۳-۱)$$

(۳-۲-۲-۲). کمينه سازي هزينه

کاهش حرکات بين سلولي از طريق تکرار ماشين هاي گلوگاهي و/ يا مقاطعه فرعي قطعات استثنايي هزينه هاي سيستم توليد را افزايش مي دهد. بديهي است از ميان دو طرح تکرار ماشين ها – مقاطعه فرعي قطعات که ميزان حرکات بين سلولي را به يک اندازه کاهش مي دهند، آن که هزينه کمتري دارد

[۱] Job Shop

ارجح است. بنابراين كمينه سازي مجموع هزينه هاي تكرار ماشين هاي گلوگاهي و مقاطعه فرعي قطعات استثنايي در قالب تابع هدف زير به عنوان يكي ديگر از اهداف مدل در نظر گرفته شده است:

$$Min \quad f_{۲} = \sum_{k} \left(\sum_{j \in EP_k} D_j . S_j . X_j + \sum_{i \in EM_k} M_i . Y_{ik} \right) \qquad (۳-۲)$$

لازم به يادآوري است كه در صورت انتخاب سياست مقاطعه فرعي براي يک قطعه، كل تقاضاي آن از طريق مقاطعه كاران تامين خواهد شد.

(۳-۲-۲-۳). كمينه سازي عدم بهره گيري از ماشين آلات[۱]

از آنجا كه سيستم سلولي اوليه از آرايش مجدد يک سيستم كارگاهي شكل گرفته و در آن سيستم كارگاهي ظرفيت كافي براي توليد همه قطعات وجود داشته بود، لذا تكرار ماشين هاي گلو گاهي و مقاطعه فرعي قطعات استثنايي موجب كاهش نسبت بهره گيري از ماشين آلات[۲] مي شود. بنابر اين ضروري است كه در تكرار ماشين آلات گلوگاهي و مقاطعه فرعي قطعات استثنايي به دنبال طرحهايي بود كه ضمن تامين اهداف ديگر، كمترين تاثير منفي بر روي اين نسبت كه يكي از شاخص هاي مهم در سود دهي بلند مدت شركت هاي توليدي است را داشته باشند. در نتيجه كمينه سازي نسبت عدم بهره گيري از ماشين آلات (كه معادل بيشينه سازي نسبت بهره گيري است) به عنوان هدف بعدي مدل منظور شده است:

$$Min \quad f_{۳} = ۱ - OU = ۱ - \frac{\sum_{k=۱}^{c} UC_k . \left(CS_k + \sum_{i \in BM_k} Y_{ik} \right)}{\sum_{k=۱}^{c} \left(CS_k + \sum_{i \in BM_k} Y_{ik} \right)} \qquad (۳-۳)$$

كه در آن:

[۱] Underutilization
[۲] Machine Utilization

$$UC_k = \frac{\sum_{i \in MC_k} \left(\sum_{j \in HF_k} D_j.t_{ij} - \sum_{j \in EP_k} D_j.t_{ij}.X_j + \sum_{j \in GF_k} D_j.t_{ij}.(1 - X_j) \right)}{\sum_{i \in MC_k} CM_i + \sum_{i \in BM_k} Y_{ik}.CM_i} \qquad (۳-٤)$$

نسبت بهره گيري از ماشين آلات در سلول k و OU نسبت بهره گيري كلي ماشين آلات مورد نظر در سيستم سلولي مي باشد. نسبت بهره گيري در هر سلول از تقسيم كل زمان ماشينكاري بر مجموع ظرفيت ماشينكاري آن سلول محاسبه مي شود. نسبت بهره گيري كلي ماشين آلات در سيستم نيز معادل ميانگين وزني نسبت بهره گيري سلولها در نظر گرفته مي شود كه در آن وزن هر سلول، تعداد ماشينهاي تخصيص يافته به آن است.

(۳-۲-۲-۴). كمينه سازي عدم تعادل بار كاري سلولها

وجود اختلاف قابل ملاحظه در بار كاري سلولها مي تواند منشا مشكلات زيادي در امر مديريت سيستم توليد به ويژه در بعد نيروي انساني آن باشد. اين مسئله مي تواند اختلافات زيادي را در ميزان نياز به اضافه كاري در سلول هاي مختلف و درآمد پرسنل شاغل در آنها ايجاد نمايد بنابراين كاهش ميزان اختلاف بار كاري سلول هاي مختلف در قالب كمينه سازي واريانس نسبت بهره گيري ماشين آلات در سلولهاي مختلف به عنوان هدف ديگر مدل طراحي سيستم سلولي در نظر گرفته شده است:

$$Min \quad f_٤ = \frac{\sum_{k=1}^{c} (UC_k - OU)^2}{c - 1} \qquad (۳-۵)$$

(۳-۳). محدوديت ها

(۳-۳-۱). محدوديت هاي اصلي

محدوديت هاي اصلي مدل متشكل از دو دسته: يكي محدوديت بزرگي سلول ها و ديگري صفر و يک بودن متغيرهاي تصميم مي باشند. واضح است كه تكرار ماشين هاي گلوگاهي در يک سلول منجر به بزرگي اندازه آن مي شود و از آنجا كه براي بزرگي سلولهاي توليدي حدود تجربي وجود دارد (باربيج ۱۹۷۵ [۹]) لذا لازم است حد بالاي بزرگي سلول را از طريق اعمال سري محدوديت هاي زير كنترل نمود:

$$(CS_k + \sum_{i \in BM_k} Y_{ik}) \le MCS \quad , \quad k = ۱,...,c \qquad (۳-۶)$$

و بالاخره متغير هاي تصميم محدود به گرفتن يكي از مقادير صفر يا يک مي باشند:

$$X_j, Y_{ik} \in \{۰,۱\} \quad , \quad \forall \ i,j,k \qquad (۳-۷)$$

(۲-۳-۳). محدوديت هاي اختياري

در صورتي كه در مسئله خاصي اطلاعاتي از ترجيحات تصميم گير روي حدود پايين قابل پذيرش در معيارهاي چهارگانه مدل در اختيار باشد مي توان با استفاده از آن فضاي شدني مسئله را محدودتر كرده و از اين طريق حجم محاسبات را تقليل داد. بدين منظور در رابطه با هر يك از معيارها يك محدوديت به شكل زير به مجموعه محدوديت هاي مدل اضافه مي شود:

$$f_i \ge \mathrm{LB_i} \ , \ i = ۱,...,٤ \qquad (۳-۸)$$

كه در آن f_i تابع هدف i و $\mathrm{LB_i}$ حد پايين مورد پذيرش روي آن مي باشند.

(۳-۴). پيچيدگي مدل

وجود هر قطعه استثنايي يک متغير تصميم و هر ماشين گلوگاهي حداقل يک متغير تصميم به مسئله اضافه مي كند.در مورد ماشين هاي گلوگاهي تعداد متغيرهاي تصميم تحت تاثير چگونگي نياز ساير سلولها به ماشين گلوگاهي قرار داشته و لذا وجود يک ماشين گلوگاهي مي تواند موجد بيش از يک متغير تصميم باشد. بنابراين بزرگي مسئله و محاسبات مورد نياز آن تابعي از تعداد قطعات استثنايي و ماشين هاي گلوگاهي يا $O\left(۲^{(N_{ep} + f(N_{bm}))}\right)$ مي باشد كه در آن N_{ep} تعداد قطعات استثنايي و N_{bm} تعدا ماشين هاي گلوگاهي هستند. $f(N_{bm})$ تابعي از تعداد ماشين هاي گلوگاهي است كه مقدار دقيق آن وابسته به تعداد سلولهايي است كه به ماشين هاي گلوگاهي نياز دارند. هر چه تعداد اين سلولها بيشتر باشد مقدار f بزرگتر خواهد بود.

(۵-۳). مثال

در یک سیستم کارگاهي متشکل از ۴ ماشین، ۶ نوع قطعه توليد مي شود. مديريت کارخانه تصميم به تغيير اين سيستم به سيستم توليد سلولي را گرفته است. به منظور شناسايي گروههاي ماشيني و خانواده قطعات ماتريس تلاقي قطعه – ماشين با استفاده از یک روش سورت ماتريسي مرتب شده است. ماتريس هاي اوليه و مرتب شده قطعه _ ماشين در شکل (۲-۳) نشان داده شده است.

		Parts					
		۱	۲	۳	٤	٥	٦
Machines	۱	۱		۱			۱
	۲	۱		۱	۱		۱
	۳		۱		۱	۱	
	٤	۱	۱		۱	۱	

(الف). ماتريس اوليه

		Parts					
		۱	۳	٦	۲	٤	٥
Machines	۲	۱	۱	۱		۱	
	۱	۱	۱	۱			
	٤	۱			۱	۱	۱
	۳				۱	۱	۱

(ب). ماتريس مرتب شده و طرح تجزيه پيشنهادي

شکل (۲-۳). ماتريسهاي اوليه و مرتب شده قطعه _ ماشين

ساير داده هاي مسئله در شکل (۳-۳) ارائه شده است. اعداد داخل ماتريس تلاقي قطعه _ ماشين شکل (۳-۳) نماينده مقادير t_{ij} مي باشند. تقاضاي ساليانه قطعات (D_j ها)، هزينه افزايشي مقاطعه فرعي قطعات (S_j ها)، و هزينه ساليانه ناشي از بکارگيري ماشين هاي گلوگاهي (M_i ها) ساير داده هاي ارائه شده در شکل (۳-۳) هستند. به علاوه فرض مي شود ظرفيت ماشينکاري تمام قطعات مساوي و معادل ۲۱۰۰۰۰ دقيقه در سال مي باشد (CM_i = ۲۱۰۰۰۰ به ازاي $i=1,...,4$). به عنوان حد بالاي اندازه سلول نيز ۳ = MCS در نظر گرفته مي شود. طرح تفکيک ارائه شده در شکل (۲-۳) تشکيل دو سلول توليدي را پيشنهاد مي نمايد. اطلاعات اين سلولها ،خانواده هاي قطعه متناظر با آنها و ساير مجموعه هاي مورد نياز در فرمول بندي مسئله در قالب مدل ٤ هدفي در جدول (۱-۳) ذکر شده است.

		Parts						
		۱	۲	۳	٤	٥	٦	M_i
Machines	۱	۸		۷			۱	۱٤،۰۰۰
	۲	۷		۳	۲		۸	٦،۰۰۰
	۳		۱۰		٦	٥		۱۹،۰۰۰
	٤	۹	۳		۱	۹		۷،۰۰۰
	D_j	۱۰،۰۰۰	۷،۰۰۰	۸،۰۰۰	۱،۰۰۰	٤،۰۰۰	۲،۰۰۰	
	S_j	۳	۳	۱	٤	۲	۳	

شکل (۳-۳). زمانهاي پردازش (t_{ij} ها) و ساير داده هاي مسئله

سلول ۱	سلول ۲
$HF_۱ = \{P۱, P۳, P٦\}$	$HF_۲ = \{P۲, P٤, P٥\}$
$MC_۱ = \{M۲, M۱\}$	$MC_۲ = \{M٤, M۳\}$
$GF_۱ = \{P٤\}$	$GF_۲ = \{P۱\}$
$BM_۱ = \{M٤\}$	$BM_۲ = \{M۲\}$
$EP_۱ = \{P۱\}$ $EM_۱ = \{M٤\}$	$EP_۲ = \{P٤\}$ $EM_٤ = \{M۲\}$
$CS_۱ = ۲$	$CS_۲ = ۲$
$PM_{۱٤} = D_۱ = ٥۰۰۰$	$PM_{٤۲} = D_٤ = ۱۰۰۰$

جدول (۱-۳). اطلاعات مجموعه هاي مسئله

در اين مسئله ۴ متغير تصميم: $X_۱$ ، $X_٤$ ، $Y_{٤,۱}$ و $Y_{۲,۲}$ وجود دارند. توابع هدف و محدوديت هاي مسئله به شرح زير مي باشند:

$$Min\ \ f_۱ = \left((۱۰،۰۰۰)(۱ - X_۱)(۱ - Y_{٤,۱})) + ((۱،۰۰۰)(۱ - X_٤)(۱ - Y_{۲,۲})\right) \qquad (۳-۸)$$

$$Min\ \ f_۲ = \begin{pmatrix} (۱۰,۰۰۰)(۳)(X_۱) + (۷,۰۰۰)(Y_{٤,۱})) + \\ ((۱,۰۰۰)(٤)(X_٤) + (٦,۰۰۰)(Y_{۲,۲}) \end{pmatrix} \qquad (۳-۹)$$

$$Min\ \ f_۳ = ۱ - OU \qquad (۳-۱۰)$$

$$Min \;\; f_{٤} = \left((UC_{١} - OU)^{٢} + (UC_{٢} - OU)^{٢}\right) \qquad (۳-۱۱)$$

که در آن:

$$OU = \frac{\left((UC_{١})(٢ + Y_{٤,١}) + (UC_{٢})(٢ + Y_{٢,٢})\right)}{\left((٢ + Y_{٤,١}) + (٢ + Y_{٢,٢})\right)} \qquad (۳-۱۲)$$

$$UC_{١} = \frac{\left((١٠,٠٠٠)(٧ + ٨) + (٨,٠٠٠)(٣ + ٧) + (٢,٠٠٠)(٨ + ١) - (١٠,٠٠٠)(٧ + ٨)(X_{١})\right)}{\left(٢١٠,٠٠٠ + ٢١٠,٠٠٠ + ٢١٠,٠٠٠(Y_{٤,١})\right)} \qquad (۳-۱۳)$$

$$UC_{٢} = \frac{\left((٧,٠٠٠)(١٠ + ٣) + (١٠,٠٠٠)(١ + ٦) + (٤,٠٠٠)(٩ + ٥) - (١٠,٠٠٠)(١ + ٦)(X_{٤})\right)}{\left(٢١٠,٠٠٠ + ٢١٠,٠٠٠ + ٢١٠,٠٠٠(Y_{٢,٢})\right)} \qquad (۳-۱٤)$$

$S.T:$

$$\left(٢ + Y_{٤,١}\right) \leq ٣ \qquad (۳-۱۵)$$

$$\left(٢ + Y_{٢,٢}\right) \leq ٣ \qquad (۳-۱۶)$$

$$X_{١}, X_{٤}, Y_{٤,١}, Y_{٢,٢} \in \{٠،١\} \qquad (۳-۱۷)$$

در جدول (۲-۳) کليه بردارهاي جواب هاي ممکن به همراه مقادير توابع هدف ۴ گانه متناظر با هر بردار محاسبه و ارائه شده است. در اين جدول همچنين نسبت بهره گيري از سلولها (UC_i ها) و نيز نسبت بهره گيري کلي سيستم سلولي (OU) که در محاسبه مقادير $f_{٣}$ و $f_{٤}$ مورد نياز هستند، به عنوان اطلاعات جنبي ارائه شده اند. از مقايسه دو به دوي بردارهاي جواب ۱۶ گانه با همديگر مي شود که جواب ۵ بهتر از جواب ۲ است چرا که مقدار $f_{١}$ مساوي و مقادير $f_{٢}$ ، $f_{٣}$ و $f_{٤}$ کمتري نسبت به جواب ۲ دارد. جواب ۲ نيز به نوبه خود بهتر از جواب ۶ است. همچنين جواب ۳ بهتر از جواب ۱۱ بوده و جواب ۴ نيز بهتر از جواب هاي ۸ ، ۹ ، ۱۲ ، ۱۳ ، ۱۵ و ۱۶ مي باشد. جواب هايي که جوابي بهتر از آنها در ميان ساير بردارهاي جواب وجود دارد به عنوان جواب هاي مغلوب[۱] با زمينه

[۱] Dominated Solutions

شماره جواب	متغيرهاي تصميم				مقادير توابع هدف				اطلاعات جنبي		
	X_1	X_4	$Y_{4,1}$	$Y_{2,2}$	f_1	f_2	f_3	f_4	UC_1	UC_2	OU
۱	۰	۰	۰	۰	۱۱۰۰۰	۰	۰,۴۱۱۹۰	۰,۰۰۰۱۰	۰,۵۹۵۲۴	۰,۵۸۰۹۵	۰,۵۸۸۱۰
۲	۰	۰	۰	۱	۱۰۰۰۰	۶۰۰۰	۰,۵۲۹۵۲	۰,۰۲۲۴۸	۰,۵۹۵۲۴	۰,۳۸۷۳۰	۰,۴۷۰۴۸
۳	۰	۰	۱	۰	۱۰۰۰	۷۰۰۰	۰,۵۲۹۵۲	۰,۰۱۷۶۳	۰,۳۹۶۸۳	۰,۵۸۰۹۵	۰,۴۷۰۴۸
۴	۰	۰	۱	۱	۰	۱۳۰۰۰	۰,۶۰۷۹۴	۰,۰۰۰۰۵	۰,۳۹۶۸۳	۰,۳۸۷۳۰	۰,۳۹۲۰۶
۵	۰	۱	۰	۰	۱۰۰۰۰	۴۰۰۰	۰,۴۲۲۶۲	۰,۰۰۰۳۴	۰,۵۹۰۴۸	۰,۵۶۴۲۹	۰,۵۷۷۳۸
۶	۰	۱	۰	۱	۱۰۰۰۰	۱۰۰۰۰	۰,۵۳۸۱۰	۰,۰۲۳۸۸	۰,۵۹۰۴۸	۰,۳۷۶۱۹	۰,۴۶۱۹۰
۷	۰	۱	۱	۰	۰	۱۱۰۰۰	۰,۵۳۸۱۰	۰,۰۱۵۱۴	۰,۳۹۳۶۵	۰,۵۶۴۲۹	۰,۴۶۱۹۰
۸	۰	۱	۱	۱	۰	۱۷۰۰۰	۰,۶۱۵۰۸	۰,۰۰۰۱۵	۰,۳۹۳۶۵	۰,۳۷۶۱۹	۰,۳۸۴۹۲
۹	۱	۰	۰	۰	۱۰۰۰	۳۰۰۰۰	۰,۶۹۷۶۲	۰,۰۰۸۲۷	۰,۲۳۸۱۰	۰,۳۶۶۶۷	۰,۳۰۲۳۸
۱۰	۱	۰	۰	۱	۰	۳۶۰۰۰	۰,۷۵۸۱۰	۰,۰۰۰۰۲	۰,۲۳۸۱۰	۰,۲۴۴۴۴	۰,۲۴۱۹۰
۱۱	۱	۰	۱	۰	۱۰۰۰	۳۷۰۰۰	۰,۷۵۸۱۰	۰,۰۲۲۴۸	۰,۱۵۸۷۳	۰,۳۶۶۶۷	۰,۲۴۱۹۰
۱۲	۱	۰	۱	۱	۰	۴۳۰۰۰	۰,۷۹۸۴۱	۰,۰۰۳۶۷	۰,۱۵۸۷۳	۰,۲۴۴۴۴	۰,۲۰۱۵۹
۱۳	۱	۱	۰	۰	۰	۳۴۰۰۰	۰,۷۰۸۳۳	۰,۰۰۶۸۱	۰,۲۳۳۳۳	۰,۳۵۰۰۰	۰,۲۹۱۶۷
۱۴	۱	۱	۰	۱	۰	۴۰۰۰۰	۰,۷۶۶۶۷	۰,۰۰۰۰۰	۰,۲۳۳۳۳	۰,۲۳۳۳۳	۰,۲۳۳۳۳
۱۵	۱	۱	۱	۰	۰	۴۱۰۰۰	۰,۷۶۶۶۷	۰,۰۱۹۶۶	۰,۱۵۵۵۶	۰,۳۵۰۰۰	۰,۲۳۳۳۳
۱۶	۱	۱	۱	۱	۰	۴۷۰۰۰	۰,۸۰۵۵۶	۰,۰۰۳۰۲	۰,۱۵۵۵۶	۰,۲۳۳۳۳	۰,۱۹۴۴۴

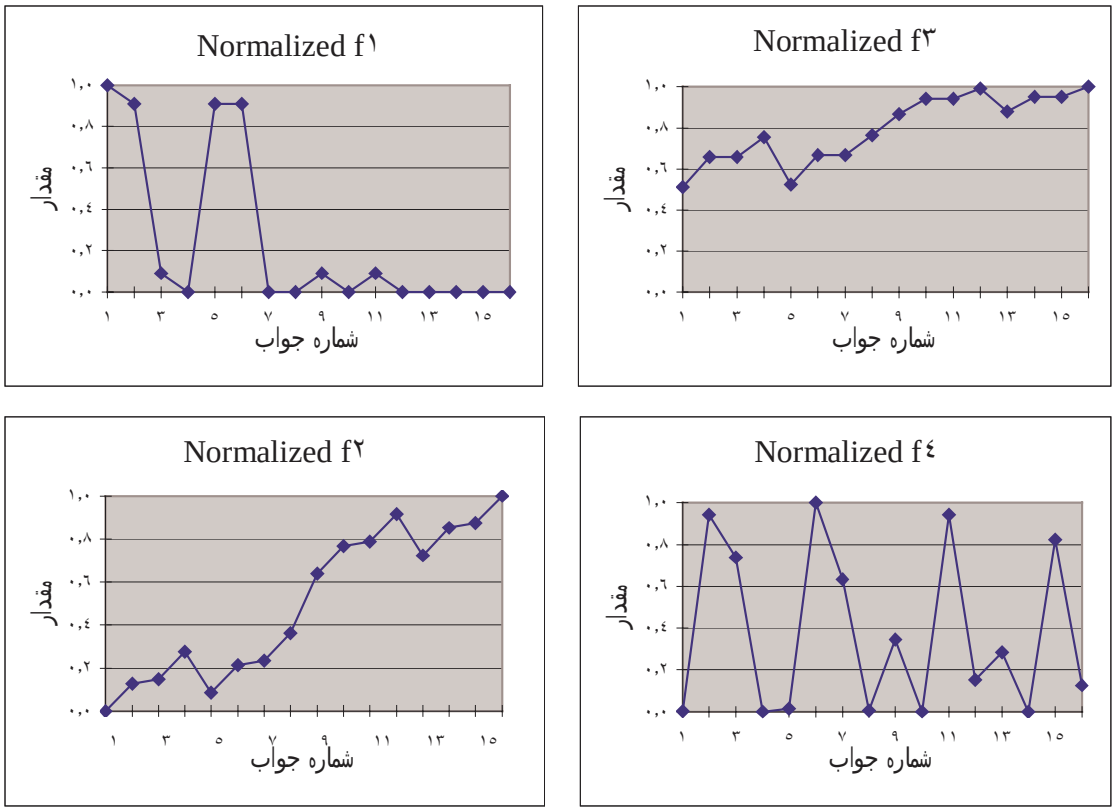
جدول (۲-۳). بردارهاي جواب ممكن و مقادير توابع هدف آنها (▤ : جواب مغلوب)

خاكستري در جدول (۲-۳) مشخص شده اند. در شكل (٤-۳) چگونگي تغييرات مقادير نرماليزه شده توابع هدف در بردارهاي جواب مختلف به تصوير كشيده شده است. اين تصاوير وجود تعارض[1] ميان اهداف ۴ گانه را نشان مي دهد به گونه اي كه نمي توان جواب واحدي كه همزمان تمام توابع هدف را كمينه سازد پيدا نمود. در نهايت، جواب هاي ۱ ، ۳ ، ۴ ، ۵ ، ۷ ، ۱۰ و ۱۴ به عنوان جواب هاي نامغلوب[2] در اختيار تصميم گير كه همان طراح سيستم است قرار مي گيرد تا از ميان آنها بر اساس ترجيحات خود انتخاب نمايد. بديهي است كه تصميم گيران مختلف ممكن است جواب هاي متفاوتي را ترجيح دهند.

(۶-۳). جمع بندي

در اين فصل مسئله چگونگي برخورد با عناصر استثنايي در طراحي يک سيستم توليد سلولي مورد بحث قرار گرفت. تكرار ماشين هاي گلوگاهي و مقاطعه فرعي قطعات استثنايي به عنوان دو سياست

[1] Conflict
[2] Non-Dominated Solutions



شکل (٤-۳). چگونگي تغييرات مقادير نرماليزه شده توابع هدف

پايه اي براي كاهش وابستگي سلولهاي توليدي به يكديگر معرفي شدند. اگرچه كاهش وابستگي سلولها موجب كاهش حركات بين سلولي به عنوان يكي از معيارهاي مهم در طراحي سيستم توليد سلولي مي شود، بر روي تعدادي از پارامترهاي سيستم از جمله هزينه تبديل سيستم و نسبت بهره گيري ماشين آلات همراه با تعادل بار كاري سلولهاي مختلف تاثير منفي مي گذارد. براي رفع اين مشكل و ايجاد بستري منطقي براي تصميم گيري در خصوص بهترين استراتژي با توجه به معيارهاي مختلف، يک مدل ۴ هدفي براي پيدا كردن جواب هاي نامغلوب مسئله طراحي سيستم توليد سلولي ارائه شد. نحوه استفاده از مدل با استفاده از يک مثال كوچک با ۴ متغير تصميم توصيف شد. حجم محاسبات مورد نياز حتي در اين مثال كوچک نيز قابل توجه بود. با توجه به وجود تعداد زياد عناصر استثنايي در مسائل واقعي توسعه و استفاده از روش هاي موثر كه قادر باشند مجموعه جواب هاي نامغلوب مسئله را در زمان محاسباتي قابل قبولي پيدا نمايند ضروري مي نمايد.

فصل ۴

مروري بر مسئله بهينه سازي چند هدفي و چگونگي استفاده از الگوريتم هاي ژنتيك و شبكه هاي عصبي در حل آن

(۱-۴). خانواده مسائل چند معياري

معيارها[۱] عبارتند از قواعد و استانداردهايي كه فرآيند تصميم گيري را هدايت مي كنند. از آنجا كه تصميم گيري از طريق انتخاب و يا فرموله كردن ويژگي ها[۲]، اهداف[۳] و آرمان ها[۴] صورت مي پذيرد، مي توان واژه معيار را به هر سه اين مفاهيم اطلاق نمود (زلني ۱۹۸۲ [۹۰]). ويژگي ها توصيف كننده موجوديت ها مي باشند نظير قيمت و سرعت كه از جمله ويژگي هاي يک خودرو محسوب مي شوند. اهداف، نماينده جهت هاي بهبود يا كارآيي ويژگي ها به صورت انفرادي و يا تركيبي مي باشند. در اين رابطه دو جهت وجود دارد: بيشتر يا كمتر، كه به ترتيب نماينده بيشينه سازي و كمينه سازي هستند. در نتيجه قيمت به خودي خود يكي از ويژگي هاي يک خودرو است اما پيدا كردن ارزان ترين خودرو از ميان گزينه هاي موجود يک هدف مي باشد. آرمان ها به طور كامل بر اساس نيازها و خواسته هاي يک تصميم گير قابل شناسايي مي باشند. آنها مقادير و يا سطوح از پيش تعيين شده اي روي ويژگي ها و

---

[۱] Criteria
[۲] Attributes
[۳] Objectives
[۴] Goals

اهداف مي باشند. به عنوان مثال اجراي يک برنامه توليد در کوتاهترين زمان ممکن يک هدف، و انجام آن در ۵ روز يک آرمان است. مسائل تصميم گيري چند معياري[۱] يا MCDM بر اساس تعاريف فوق توسط هوانگ و يون (۱۹۸۱) به صورت زير دسته بندي شده است [۳٦]:

- تصميم گيري بر اساس ويژگي هاي چندگانه[۲] يا MADM
- تصميم گيري بر اساس اهداف چندگانه[۳] يا MODM

MADM عمدتا" به مسائل انتخاب از ميان تعدادي گزينه محدود مي پردازد حال آنکه MODM با گزينه هاي بسيار زياد (احتمالا" نامحدود) سر و کار دارد و بيشتر در مسائل طراحي به کار گرفته مي شود. هوانگ و يون (۱۹۸۱) مقايسه تفصيلي تري ما بين اين دو خانواده ارائه کرده اند [۳٦]. جدول (۱-۴) نتيجه اين مقايسه را نشان مي دهد.

مبناي مقايسه	MADM	MODM
معيار مبناي تصميم گيري	ويژگي ها	اهداف
چگونگي تعريف اهداف	تلويحي	صريح و روشن
چگونگي تعريف ويژگي ها	صريح و روشن	تلويحي
محدوديت ها	غير فعال (در ويژگي ها منظور شده اند)	فعال
گزينه ها	محدود، گسسته	نامحدود، پيوسته
تعامل با تصميم گير	خيلي زياد نيست	غالبا" وجود دارد
مورد استفاده	انتخاب، ارزيابي	طراحي

جدول (۱-۴). مقايسه MADM و MODM

مسئله طراحي سلول هاي توليدي با توجه به معيارهاي چندگانه که در آن تعداد جواب هاي شدني بسيار زياد مي باشد با مسائل MODM سازگاري دارد. در ادامه توضيحات بيشتري در مورد اين نوع مسائل ارائه خواهد شد.

[۱] Multi Criteria Decision Making
[۲] Multi Attribute Decision Making
[۳] Multi Objective Decision Making

(۲-۴). بهينه سازي چند هدفي

مسائل بهينه سازي چند هدفي[۱] يا MOP کاربردهاي مختلفي در مسائل مهندسي دارند. در بسياري از مسائل طراحي، کمينه سازي هزينه و بيشينه سازي کارآيي ، قابليت اتکا و عوامل ديگر از جمله اهداف مورد نظر طراح مي باشند. اين معيارها ممکن است به صورت صريح و در قالب توابع هدف ذکر شوند و يا به صورت ضمني به شکل محدوديت در مدل منظور گردند. مسئله بهينه سازي چند هدفي را مي توان به صورت زير تعريف نمود:

تعريف ۱. مسئله بهينه سازي چند هدفي

مسئله MOP را در حالت کلي مي توان به زبان رياضي به شکل زير بيان کرد:

$$\begin{aligned} &Max \quad \boldsymbol{y} = \boldsymbol{f(x)} = (f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_k(\boldsymbol{x})) \\ &S.T: \quad \boldsymbol{e}(\boldsymbol{x}) = (e_1(\boldsymbol{x}), e_2(\boldsymbol{x}), \ldots, e_m(\boldsymbol{x})) \leq 0 \end{aligned} \tag{۴-۱}$$

که در آن:

$$\boldsymbol{x} = (x_1, x_2 \ldots, x_n) \in \boldsymbol{X}$$
$$\boldsymbol{y} = (y_1, y_2 \ldots, y_n) \in \boldsymbol{Y}$$

و $\boldsymbol{x}$ بردار تصميم[۲]، $\boldsymbol{y}$ بردار هدف[۳]، $\boldsymbol{X}$ فضاي تصميم[۴] و $\boldsymbol{Y}$ فضاي هدف[۵] هستند. محدوديت هاي $\boldsymbol{e}(\boldsymbol{x}) \leq 0$ مجموعه جواب هاي شدني[۶] را مشخص مي کنند.

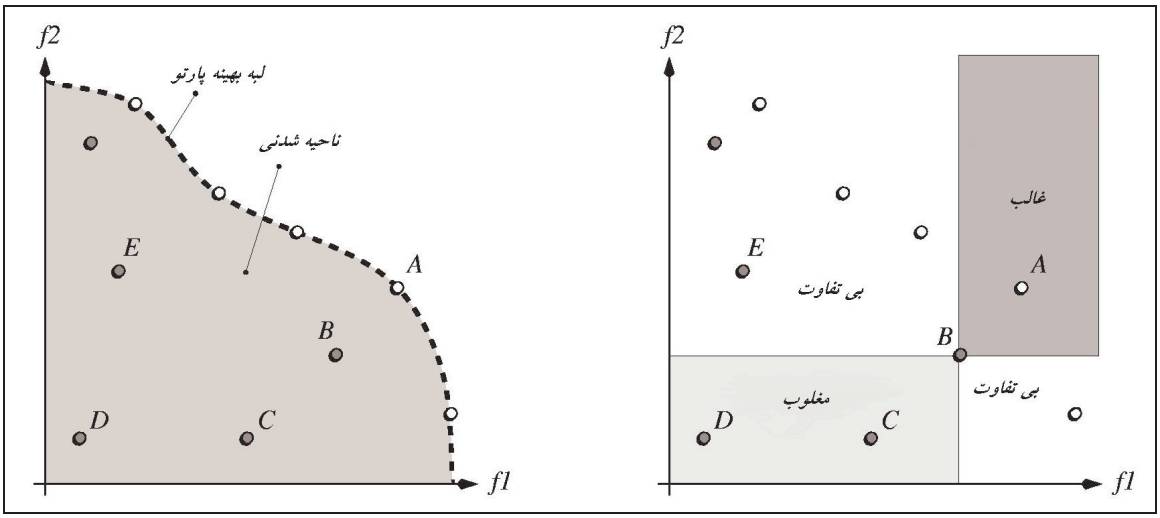
تعريف ۲. مجموعه شدني

مجموعه شدني $\boldsymbol{X}_f$ مشتمل بر جواب هاي $\boldsymbol{x}$ است که محدوديت هاي $\boldsymbol{e}(\boldsymbol{x})$ را ارضا مي نمايند، يعني:

$$\boldsymbol{X}_f = \{\boldsymbol{x} \in \boldsymbol{X} \mid \boldsymbol{e}(\boldsymbol{x}) \leq 0\} \tag{۴-۲}$$

---

[۱] Multiobjective Optimization Problems
[۲] Decision Vector
[۳] Objective Vector
[۴] Decision Space
[۵] Objective Space
[۶] Feasible Solutions



شکل (۱-۴). بيان تصويري بهينگي پارتو در فضاي هدف (چپ) و روابط ممکن بين جواب ها در فضاي هدف (راست)

تصوير X_f يا همان ناحيه شدني در فضاي هدف به صورت $Y_f = f(X_f) = \bigcup_{x \in X_f} \{f(x)\}$ تعريف مي شود. در حالتي که توابع هدف از نوع کمينه سازي[۱] باشند نيز مي توان تعاريف مشابهي را ارائه نمود. مشکل عمده در حل مسائل بهينه سازي چند هدفي از آنجا ناشي مي شود که جواب بهينه شدني توابع هدف مختلف اين مسئله لزوما" با هم يکي نبوده و حتي در بسياري موارد متعارض[۲] يکديگر نيز مي باشند. در چنين شرايطي نمي توان همه اهداف را به صورت همزمان بهينه نمود. در عوض بايد به جستجوي تعادل رضايت بخشي[۳] مابين اين جواب ها پرداخت. به همين لحاظ مفاهيم بهينگي ويژه اي در مسائل بهينه سازي چند هدفي مورد نياز مي باشد. در مسائل بهينه سازي تک هدفي[٤] يا SOP مجموعه جواب هاي شدني به طور کامل بر اساس تابع هدف f مرتب شده اند به اين صورت که در مورد دو جواب $a, b \in X_f$ يا $f(a) \geq f(b)$ و يا $f(b) \geq f(a)$. در اينجا هدف پيدا کردن جواب يا جواب هايي است که تابع f را بيشينه سازند. هنگامي که مسئله بيش از يک هدف داشته باشد، X_f مجموعه کاملا" مرتب[۵] نيست بلکه در حالت کلي يک مجموعه مرتب جزئي[٦] است (پارتو ۱۸۹۶ [٦۱]). اين مفهوم در سمت چپ شکل (۱-۴) نشان داده شده است. اين شکل

[۱] Minimization
[۲] Conflicting
[۳] Satisfactory Trade-Off
[٤] Single-objective Optimization Problem
[۵] Totally Ordered
[٦] Partially Ordered

مربوط به مسئله بيشينه سازي همزمان دو هدف f_1 و f_2 است. در اينجا جواب B نسبت به جواب C ارجح است چرا که همزمان مقادير f_1 و f_2 بزرگتري دارد. همچنين C نيز بر D ارجحيت دارد چرا که در عين داشتن f_2 مساوي، f_1 بزرگتري دارد. براي بيان رياضي اين وضعيت، روابط $=$ ، $\geq$ و $\leq$ در تناظر با حالت تک هدفي براي بردارهاي هدف نيز توسعه يافته اند .

تعريف ۳. به ازاي هر دو بردار هدف $\boldsymbol{u}$ و $\boldsymbol{v}$:

$$\begin{aligned} \boldsymbol{u} &= \boldsymbol{v} \quad iff \quad \forall i \in \{1, 2, \ldots, k\} : u_i = v_i \\ \boldsymbol{u} &\geq \boldsymbol{v} \quad iff \quad \forall i \in \{1, 2, \ldots, k\} : u_i \geq v_i \\ \boldsymbol{u} &> \boldsymbol{v} \quad iff \quad u_i \geq v_i \ \wedge \ u_i \neq v_i \end{aligned} \tag{۴-۳}$$

روابط $>$ و $\leq$ نيز به طريق مشابه تعريف مي شوند.

با استفاده از اين تعاريف اگر $\mathbf{B} > \mathrm{C}$ و $\mathbf{C} > \mathbf{D}$ در نتيجه $\boldsymbol{B} > \boldsymbol{D}$ خواهد بود. بردارهاي تصميم $\boldsymbol{a}$ و $\boldsymbol{b}$ بر اساس رابطه $\geq$ در MOP مي توانند يکي از سه وضعيت: $\boldsymbol{f}(\boldsymbol{a}) \geq \boldsymbol{f}(\boldsymbol{b})$ ، $\boldsymbol{f}(\boldsymbol{b}) \geq \boldsymbol{f}(\boldsymbol{a})$ يا $\boldsymbol{f}(\boldsymbol{a}) \ngeq \boldsymbol{f}(\boldsymbol{b}) \wedge \boldsymbol{f}(\boldsymbol{b}) \ngeq \boldsymbol{f}(\boldsymbol{a})$. بنابراين نمادها و عبارت هاي زير براي دسته بندي حالت هاي مختلف مورد استفاده قرار گرفته است.

تعريف ۴. غلبه (چيرگي) پارتو[1]

براي هر دو بردار تصميم $\boldsymbol{a}$ و $\boldsymbol{b}$:

$$\begin{aligned} \boldsymbol{a} &\succ \boldsymbol{b} \ (\text{غلبه دارد}^{2}\ \boldsymbol{b}\ \text{بر}\ \boldsymbol{a}) && iff \quad \boldsymbol{f}(\boldsymbol{a}) > \boldsymbol{f}(\boldsymbol{b}) \\ \boldsymbol{a} &\succeq \boldsymbol{b} \ (\text{غلبه ضعيف دارد}^{3}\ \boldsymbol{b}\ \text{بر}\ \boldsymbol{a}) && iff \quad \boldsymbol{f}(\boldsymbol{a}) \geq \boldsymbol{f}(\boldsymbol{b}) \\ \boldsymbol{a} &\sim \boldsymbol{b} \ (\text{بي تفاوت است}^{4}\ \boldsymbol{b}\ \text{نسبت به}\ \boldsymbol{a}) && iff \quad \boldsymbol{f}(\boldsymbol{a}) \ngeq \boldsymbol{f}(\boldsymbol{b}) \wedge \boldsymbol{f}(\boldsymbol{b}) \ngeq \boldsymbol{f}(\boldsymbol{a}) \end{aligned} \tag{۴-۴}$$

در مورد مسائل کمينه سازي روابط $(\prec, \preceq, \sim)$ به طور متناظر تعريف مي شوند.

در سمت راست شکل (۱-۴)، مستطيل خاکستري کمرنگ (پايين – چپ) نشان دهنده بردارهاي هدفي است که جواب هاي متناظر با آنها **مغلوب** B مي باشند. همچنين مستطيل خاکستري پر رنگ (بالا –

[1] Pareto Dominance
[2] **a** dominates **b**
[3] **a** weakly dominates **b**
[4] **a** is indifferent to **b**

راست) حاوي بردارهاي هدفي است که جواب هاي متناظر با آنها **غالب** بر B هستند. جوابهايي که بردارهاي هدف آنها در هيچيک از اين دو مستطيل واقع نيست، نسبت به B **بي تفاوت** مي باشند.

براساس مفهوم چيرگي پارتو مي توان معيار بهينگي در MOP را تعريف نمود. بردار هدف A در شکل (۱-۴) ويژگي منحصر به فردي نسبت به بردارهاي B، C، D و E دارد و آن اينست که بردار جواب متناظر با آن يا $\boldsymbol{a}$ مغلوب هيچ بردار جوابي نيست. بنابراين مي توان $\boldsymbol{a}$ را بهينه دانست از اين نظر که نمي توان هيچيک از اهداف آن را بهبود داد بدون آنکه مقدار هدف ديگر آن را بدتر نمود. به چنين جوابي **بهينه پارتو**[۱] يا **ناپست**[۲] مي گويند (کوهن ۱۹۷۸ [۱۷]).

تعريف ۵. بهينگي پارتو[۳]

بردار تصميم $\boldsymbol{x} \in \boldsymbol{X}_f$ نسبت به مجموعه $\boldsymbol{A} \subseteq \boldsymbol{X}_f$ *نامغلوب*[٤] خوانده مي شود، اگر و تنها اگر:

$$\nexists \ \boldsymbol{a} \in \boldsymbol{A} : \boldsymbol{a} \succ \boldsymbol{x} \qquad (۴-۵)$$

همچنين $\mathbf{x}$ *بهينه پارتو* است اگر و تنها اگر نسبت به $\boldsymbol{X}_f$ نامغلوب باشد.

نقاط سفيد در شکل (۱-۴) نماينده جواب هاي بهينه پارتو مي باشند. اين نقاط نسبت به يکديگر بي تفاوت هستند. در اينجا تفاوت اصلي مسائل چند هدفي با مسائل تک هدفي آشکار مي شود: مسائل چند هدفي محدود به يک جواب بهينه واحد وجود نيستند بلکه در آنها مجموعه اي از جواب هاي بهينه (پارتو) وجود دارد. هيچيک از اين جواب ها را نمي توان برتر از ديگري دانست مگر آنکه ترجيحات تصميم گير تعريف شده باشند.

مجموعه تمام جواب هاي بهينه پارتو در يک MOP *مجموعه بهينه پارتو*[۵]، و بردارهاي هدف متناظر با آنها *لبه يا سطح بهينه پارتو*[٦] ناميده مي شوند.

---

[۱] Pareto-Optimal
[۲] Non-Inferior
[۳] Pareto Optimality
[٤] Non-dominated
[۵] Pareto-Optimal Set
[٦] Pareto-Optimal Front (Surface)

تعريف ۶. مجموعه ها و لبه هاي نامغلوب[۱]

$A \subseteq X_f$ را در نظر بگيريد. تابع $p(A)$ مجموعه بردارهاي تصميم نامغلوب در A را توليد مي کند:

$$p(A) = \{ a \in A \mid \text{با توجه به } A \text{ نامغلوب است } a \} \qquad (۴-۶)$$

مجموعه $p(A)$ مجموعه نامغلوب با توجه به A است و مجموعه بردارهاي متناظر $f(p(A))$ لبه نامغلوب با توجه به A است. به علاوه مجموعه $X_p = p(X_f)$ مجموعه بهينه پارتو، و $Y_p = f(X_p)$ لبه بهينه پارتو ناميده مي شوند.

(۳-۴). جستجو و تصميم گيري

در حل يک MOP دو نوع پيچيدگي مفهومي را مي توان شناسايي کرد: جستجو[۲] و تصميم گيري[۳] (هورن ۱۹۹۷ [۳۲]). ويژگي اول مربوط به فرآيند بهينه سازي است که در آن مجموعه اي به عنوان نمونه اي از مجموعه بهينه پارتو شناسايي مي شود. همانند بهينه سازي تک هدفي، فضاهاي جواب بزرگ و پيچيده، فرآيند جستجو را مشکل ساخته و امکان استفاده از روشهاي بهينه سازي قطعي[۴] نظير برنامه ريزي خطي را محدود مي سازد (استوئر ۱۹۸۶ [۷۸]) .ويژگي ديگر به چگونگي انتخاب يک جواب سازشي[۵] از ميان مجموعه جواب بهينه پارتو توسط تصميم گير[۶] يا DM مي پردازد.

بسته به چگونگي ترکيب فرآيندهاي بهينه سازي و تصميم گيري، روشهاي بهينه سازي چند هدفي را مي توان به سه دسته تقسيم نمود (هوانگ و مسعود ۱۹۷۹ [۳۵]) :

- **تصميم گيري پيش از جستجو:** توابع هدف MOP در قالب يک تابع هدف واحد مجتمع مي شوند. اين تابع هدف به طور ضمني ترجيحات تصميم گير را در خود لحاظ کرده است.

---

[۱] Non-dominated Sets and Fronts
[۲] Search
[۳] Decision-Making
[۴] Exact
[۵] Compromise
[۶] Decision Maker

- **جستجو پيش از تصميم گيري:** بهينه سازي بدون هيچ گونه اطلاعاتي از ترجيحات تصميم گير انجام مي شود. نتيجه اين جستجو مجموعه جواب هاي نامزدي است (که در حالت ايده آل بهينه پارتو هستند) که انتخاب نهايي از ميان آنها توسط تصميم گير صورت مي پذيرد.
- **تصميم گيري در حين جستجو:** تصميم گير ترجيحات خود را در حين فرآيند بهينه سازي تعاملي[۱] ابراز مي دارد. پس از هر گام بهينه سازي تعدادي از جواب ها به تصميم گير ارائه مي شوند. وي سپس بر اساس اين جواب ها ترجيحات خود را مشخص مي نمايد و به اين ترتيب فرآيند جستجو هدايت مي شود.

ادغام توابع هدف چندگانه در قالب يک تابع هدف اين مزيت را دارد که مي توان در حل آن از روش هاي موجود براي حل مسائل تک هدفي استفاده نمود. البته اين کار مستلزم داشتن اطلاعات محيطي گسترده در مورد مسئله مي باشد که معمولا" از ابتدا در دسترس نمي باشند. انجام جستجو پيش از تصميم گيري اين مشکل را مرتفع مي سازد ولي در عوض از مزيت استفاده از ترجيحات تصميم گير که مي تواند فضاي جواب را کوچک کند محروم مي باشد. و بالاخره با ادغام جستجو و تصميم گيري مي توان از مزاياي هر دو دسته روش هاي فوق بهره گيري نمود.

(۴-۴). رويکردهاي سنتي

روش هاي کلاسيک پيدا کردن جواب هاي بهينه پارتو در بهينه سازي چند هدفي بر اساس ايده تصميم گيري پيش از جستجو، اهداف چندگانه را در قالب يک هدف با اعمال پارامترهايي ادغام مي کنند. با اين وجود، اين پارامترها توسط تصميم گير معين نمي شوند بلکه به طور خودکار توسط فرآيند بهينه سازي تغيير داده مي شوند. اين مسئله به ازاي هر مجموعه پارامتر حل شده و اين کار به دفعات تکرار مي شود تا جايي که تقريب خوبي از مجموعه جواب هاي بهينه پارتو بدست آيد.

نمونه هايي از اين خانواده: روش وزن دهي[۲] ، روش محدوديت[۳] و برنامه ريزي آرماني[۴] مي باشند. براي مطالعات تکميلي در اين زمينه مي توان به هوانگ و يون (۱۹۸۱) [۳۵]، هوانگ و مسعود (۱۹۷۸) [۳٦] و استوئر (۱۹۸۶) [۷۸] مراجعه نمود.

---

[۱] Interactive
[۲] Weighting method
[۳] Constraint method
[۴] Goal Programming

مزيت عمده روش هاي سنتي حل مسائل چند هدفي از طريق تبديل آنها به مسائل تک هدفي، مي تواند وجود روش هاي موثر براي حل آنها باشد. براي مسائل چند هدفي در مقياس بزرگ تا چند سال پيش روش هاي نادري در اختيار بود (هورن ۱۹۹۷ [۳۲]). در عوض طيف وسيعي از روش هاي ابتکاري براي حل مسائل بهينه سازي تک هدفي پيچيده توسعه داده شده بود، به عنوان مثال: الگوريتم هاي جستجوي تصادفي[۱] (تورن و زبلينسکاس ۱۹۸۹ [۸۲])، الگوريتم هاي تصادفي جستجوي محلي[۲] (هورست و پاردالوس ۱۹۹۵ [۳٤])، روش SA [۳] و روش TS [٤] و نظاير آنها. به علاوه در قسمت هاي قبلي محدوديت هاي زير در استفاده از روش هاي سنتي وجود دارد:

- برخي از اين روش ها مثل روش وزن دهي نسبت به شکل ناحيه بهينه پارتو حساس هستند.
- اطلاعاتي در باره مسئله و حدود جواب هاي آن مورد نياز است که ممکن است در اختيار نباشد.

دب (۱۹۹۹) مشکل ديگر روش هاي سنتي را محدود بودن دامنه کاربردي آنها مي داند [۲۱]. به علاوه يک مسئله مشترک در تمام روش هاي سنتي نياز به حل در دفعات زياد براي پيدا کردن يک تقريب خوب از مجموعه بهينه پارتو مي باشد. از آنجا که اين تکرارها مستقل از يکديگر انجام مي شوند، امکان بهره گيري از دست آوردهاي تکرارهاي مختلف از ميان رفته و بدين ترتيب زمان لازم براي محاسبات افزايش مي يابد.

(۵-۴). چگونگي استفاده از الگوريتم هاي ژنتيک و شبکه هاي عصبي

الگوريتم هاي ژنتيک و شبکه هاي عصبي دو تکنيک نسبتا" جديد هستند که از آنها در مسئله طراحي سلولهاي توليدي استفاده شده است. دو ويژگي مشترک اين دو تکنيک يکي الهام گيري آنها از پديده هاي طبيعي خلقت و ديگري قابليت پردازش موازي مي باشد. الگوريتم هاي ژنتيک با الهام از فرآيند تکامل طبيعي و به عنوان يک روش جستجوي هوشمند در حل مسائل بهينه سازي کاربرد گسترده اي

[۱] Random Search Algorithms
[۲] Stochastic Local Search Algorithms
[۳] Simulated Annealing
[٤] Tabu Search

يافته است. شبکه هاي عصبي با شبيه سازي فرآيند يادگيري مغز کاربرد وسيعي در تشخيص الگو و ساير استنتاج هاي هوشمندانه پيدا کرده است. همچنين مواردي از کاربرد شبکه هاي عصبي در حل مسائل بهينه سازي گزارش شده است.

در اين قسمت ضمن معرفي کلي اين دو تکنيک و طرز کار آنها، به بررسي قابليت آنها در حل مسئله چند هدفي طراحي سلولهاي توليدي خواهيم پرداخت.

(۱-۵-۴). الگوريتم هاي ژنتيک

(۱-۱-۵-٤). چارچوب يک الگوريتم ژنتيک استاندارد

يک الگوريتم ژنتيک [۱] يک روش جستجوي تطبيق پذير عمومي است که در حل يک مسئله نيازمند دانش محيطي آن مسئله مي باشد. اين روش که توسط هلند (۱۹۷۵) [۳۰] ارائه شد با الهام از فرآيند تکامل طبيعي و استفاده از ترکيب اصل بقاي اصلح[۲] و تبادل اطلاعات در ميان جوابها به توليد فرزندان[۳] مي پردازد. اين فرزندان در هر نسل[٤] جايگزين جوابهاي ضعيف مي شوند. تفاوت هاي اساسي الگوريتم هاي ژنتيک با روشهاي سنتي جستجو در بهينه سازي در موارد زير است:

- الگوريتم هاي ژنتيک به جاي استفاده از خود پارامترها با کدهاي آنها کار مي کنند
- الگوريتم هاي ژنتيک به جستجوي يک خانواده از نقاط مي پردازند و نه صرفا" يک نقطه
- الگوريتم هاي ژنتيک از اطلاعات تابع شايستگي[٥] استفاده مي کند و نيازي به مشتقات و يا ساير اطلاعات اضافي ندارند
- الگوريتم هاي ژنتيک از قواعد انتقال احتمالي در عوض قواعد قطعي استفاده مي کنند

توسعه يک الگوريتم ژنتيک نيازمند موارد زير است:

۱. يک نمايش کروموزومي[٦] از جواب هاي ممکن
۲. يک جامعه اوليه

---

[۱] Genetic Algorithm
[۲] Survival of the fittest
[۳] Offspring
[٤] Generation
[٥] Fitness function
[٦] Chromosomal representation

۳. يک تابع ارزيابي براي رتبه بندي جوابها بر اساس ميزان شايستگي آنها
٤. عملگرهاي ژنتيک[۱] که ترکيب اجزاي فرزندان را در حين تکثير[۲] تغيير مي دهند

در GA هاي ساده هر جواب بالقوه توسط دنباله اي از ژن ها[۳] که کروموزوم[٤] ناميده مي شود نمايش داده مي شود. پتانسيل هر کروموزوم به عنوان يک جواب توسط مقدار تابع شايستگي آن با توجه به تابع هدف مسئله بهينه سازي در دست حل معين مي شود. يک مجموعه انتخابي از کروموزوم ها يک جامعه[٥] ناميده شده و اين جامعه در يک مقطع زماني مشخص يک نسل[٦] خوانده مي شود. اندازه جامعه از نسلي به نسل بعدي ثابت مي ماند و اين اندازه تاثير قابل ملاحظه اي بر روي کارآيي GA دارد. مکانيزم GA عموما" بر پايه سه عملگر اصلي زير عمل مي کند (گلدبرگ ۱۹۸۹ [٢٤]): توليد مثل[۷] ، ترکيب[۸] و جهش[۹] .

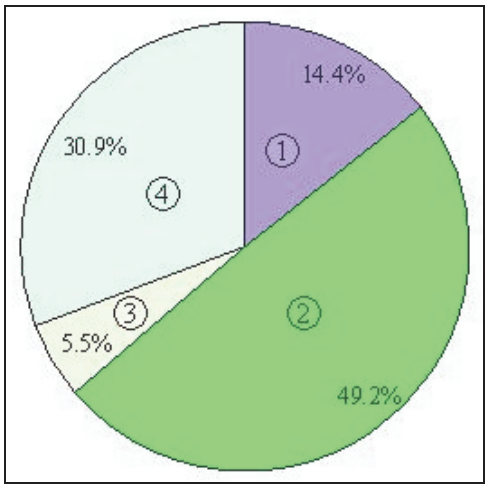
توليد مثل، فرآيندي است که در آن کروموزوم ها بر اساس مقدار تابع هدف خود (که در علم زيست شناسي تابع شايستگي خوانده مي شود) عينا" در نسل بعدي کپي مي شوند. کپي کردن کروموزوم ها بر اساس مقدار شايستگي بدين معني است که کروموزوم هاي داراي شايستگي بيشتر با احتمال بيشتري يک فرزند يا بيشتر در نسل بعدي خواهند داشت. روش هاي مختلفي براي اجراي عملگر توليد مثل وجود دارد. ساده ترين راه براي اين کار استفاده از روش چرخ رولت[۱۰] است که در آن هر کروموزوم داراي قطاعي از يک چرخ رولت فرضي متناسب با شايستگي خود است. براي توصيف بيشتر اين روش فرض کنيد در جامعه اي با ۴ کروموزوم مقادير شايستگي به شرح جدول (۲-۴) داده شده است:

---

[۱] Genetic operators
[۲] Reproduction
[۳] Genes
[٤] Chromosome
[٥] Population
[٦] Generation
[۷] Reproduction
[۸] Crossover
[۹] Mutation
[۱۰] Roulette Wheel

ردیف	کروموزوم	شایستگي	درصد از کل
۱	۰۱۱۰۱	۱۶۹	۱۴,۴
۲	۱۱۰۰۰	۵۷۶	۴۹,۲
۳	۰۱۰۰۰	۶۴	۵,۵
۴	۱۰۰۱۱	۳۶۱	۳۰,۹
	مجموع	۱۱۷۰	۱۰۰,۰

جدول (۲-۴). مثالي از یک جامعه و شایستگي اعضاي آن



شکل (۲-۴). سهم کروموزوم ها درچرخ رولت

مجموع مقادیر شایستگي ۴ کروموزوم مساوي ۱۱۷۰ است. در صد شایستگي هر کروموزوم در این مجموع در ستون آخر جدول محاسبه شده است. حال چرخ رولت را مي توان با استفاده از این درصدها، همانطور که در شکل (۲-۴) نشان داده شده، تقسیم بندي نمود. با هر بار چرخاندن این چرخ یک کروموزوم انتخاب مي شود. بدیهي است کروموزومي که شایستگي بیشتري دارد با احتمال بزرگتري انتخاب خواهد شد.

روش چرخ رولت عليرغم سادگي اجرايي، روش چندان مناسبي براي انتخاب کروموزوم ها نمي باشد. ضعف عمده اين روش مربوط به احتمال نسبتا" زياد انتخاب نشدن کروموزوم هاي با شايستگي زياد است. به عنوان نمونه در مثال بالا، کروموزوم ۲ با داشتن بالاترين شايستگي به ميزان ۴۹,۲ درصد، با احتمال ۵۰,۸ در صد در يک دور انتخاب نخواهد شد. به منظور رفع اين نقيصه روش هاي ديگري براي انتخاب کروموزوم ها ارائه شده است. براي اطلاعات بيشتر در مورد روشهاي مختلف انتخاب در الگوريتم هاي ژنتيک مي توان به گلدبرگ (۱۹۸۹) [٢٤] و بليکل و تيل (۱۹۹۵) [٦] مراجعه نمود.

پس از توليد مثل، عملگر ترکيب روي کروموزوم ها اعمال مي شود. ابتدا براي هر کروموزوم جفتي به تصادف انتخاب شده و پس از آن هر کروموزوم با جفت خود ترکيب مي شود. عملگر ترکيب ساده به اين صورت عمل مي کند: موقعيت صحيح k در طول رشته کروموزومي با احتمال يکنواخت در جايي بين ژن و آخر (موقعيت l) انتخاب مي شود. سپس ژنهاي واقع در موقعيت هاي مابين $k+۱$ تا l در دو کروموزوم با هم تعويض مي شوند. به عنوان مثال کروموزوم هاي $A_۱$ و $A_۲$ از مثال فوق را در نظر بگيريد:

$$A_۱ = ۰\ ۱\ ۱\ ۰\ |\ ۱$$
$$A_۲ = ۱\ ۱\ ۰\ ۰\ |\ ۰$$

فرض کنيد در انتخاب تصادفي يک عدد مابين ۱ تا ۵ ، موقعيت $k = ٤$ (که با علامت | مشخص شده) انتخاب شده است. به دنبال اجراي عملگر ترکيب ساده از محل انتخاب شده، ۲ کروموزوم جديد $A'_۱$ و $A'_۲$ براي نسل بعدي شکل مي گيرند:

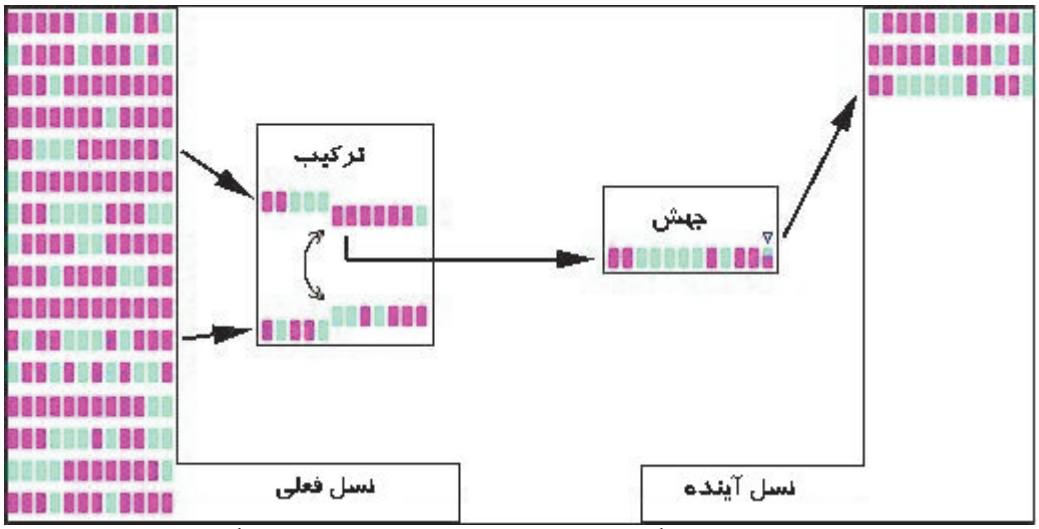
$$A'_۱ = ۰\ ۱\ ۱\ ۰\ ۰$$
$$A'_۲ = ۱\ ۱\ ۰\ ۰\ ۱$$

مکانيزم توليد مثل و ترکيب که در آن از انتخاب تصادفي و جابجايي تصادفي بخشهايي از کروموزوم ها استفاده مي شود، بخش اعظم توانمندي هاي يک الگوريتم ژنتيک را ايجاد مي کند (گلدبرگ ۱۹۸۹ [٢٤]). با اين وجود عملگر جهش براي جلوگيري از همگرايي زودرس[۱] و جستجوي بخشهاي بيشتري از فضاي جواب مورد نياز است. اين عملگر به طور تصادفي و بر اساس يک احتمال از پيش تعيين

[۱] Premature Convergence

شده، محتواي يک ژن را تغيير مي دهد. در کروموزوم هاي مثال فوق الذکر، اين کار به معني تغيير يک ۱ به ۰ و بالعکس مي باشد. شکل (۲-۵) تصويري نمادين از طرز کار انتخاب، ترکيب و جهش را نشان مي دهد.

عملگرهاي ديگري از جمله: نخبه گرايي[۱] ، معکوس سازي[۲] ، چيرگي[۳] ، تکرار درون کروموزومي[٤] و حذف[٥] نيز براي استفاده در الگوريتم هاي ژنتيک وجود دارند. همچنين مفاهيمي نظير قلمروسازي[٦] و شراکت[۷] براي بهبود کارآيي الگوريتم هاي ژنتيک در بهينه سازي توابع چند کوهانه[۸] و نيز بهينه سازي



شکل (۳-۴). طرز کار عملگرهاي انتخاب، ترکيب و جهش در الگوريتم ژنتيک

چند هدفي ارائه شده است (گلدبرگ ۱۹۸۹ [۲٤]). با اين وجود يک الگوريتم ژنتيک ساده شامل سه عملگر پايه: توليد مثل، ترکيب و جهش با يک تنظيم پارامتر مناسب مي تواند تعدادي از مسائل مهم بهينه سازي را حل کند. شکل (۴-۴) شبه کد[۹] يک الگوريتم ژنتيک استاندارد را نشان مي دهد.

---

[۱] Elitism
[۲] Inversion
[۳] Dominance
[٤] Intrachromosomal Duplication
[٥] Deletion
[٦] Niching
[۷] Sharing
[۸] Multi-modal Functions
[۹] Pseudo code

```
begin GA
   g:=۰  {generation counter}
   Initialise population p(g)
   Evaluate population p(g)

   while not done do
      g:=g+۱
      Select p(g) from p(g-۱)
      Crossover p(g)
      Mutate p(g)
      Evaluate p(g)
   end while
end GA
```

شکل (۴-۴) شبه کد يک الگوريتم ژنتيک استاندارد

(۲-۱-۵-٤). کاربرد الگوريتم هاي ژنتيک در بهينه سازي

براي حل يک مسئله بهينه سازي توسط الگوريتم هاي ژنتيک بايد بتوان فضاي جواب مسئله را با استفاده از يک نمايش کروموزومي پوشش داد. همچنين لازم است تابع هدف مسئله بهينه سازي به عنوان مبناي رتبه بندي و ارزيابي اين کروموزوم ها قرار گيرد. بر اين اساس عملا" محدوديتي بر سر راه استفاده از الگوريتم هاي ژنتيک در حل مسائل بهينه سازي وجود ندارد. دجونگ (۱۹۷۵) [۲۲] با استفاده از ۵ تابع آزمايشي به بررسي کارآيي الگوريتم هاي ژنتيک در حل مسائل بهينه سازي پرداخت. مسائل انتخابي وي به گونه اي بودند که ويژگي هاي زير را در مسائل بهينه سازي پوشش مي دادند:

- پيوستگي و ناپيوستگي
- محدب و نامحدب بودن
- تک کوهانه و چند کوهانه بودن
- کوادراتيک و غير کوادراتيک بودن
- کم بعد[۱] و پر بعد[۲] بودن

[۱] Low-Dimensionality
[۲] High-Dimensionality

- قطعي و احتمالي بودن

بر اساس تحقيقات دجونگ، الگوريتم ژنتيک در بهينه سازي توابع هموار تک کوهانه اولويت خاصي نسبت به ساير روشهاي بهينه سازي ندارد. با اين وجود در حالت کلي و در شرايطي که با توابع هدف غير هموار و چند کوهانه سر و کار داشته باشيم، الگوريتم هاي ژنتيک عملکرد خوبي در بهينه سازي آنها از خود نشان داده اند (گلدبرگ ۱۹۸۹ [٢٤]).

(۳-۱-۵-٤). چگونگي اعمال محدوديت ها

ساده ترين راه براي اعمال محدوديت هاي مسئله در الگوريتم ژنتيک، تخصيص مقادير کوچک شايستگي به جواب هاي نشدني است (گلدبرگ ۱۹۸۹ [٢٤]). بدين ترتيب جواب هاي نشدني خود به خود در فرآيند انتخاب حذف مي شوند.

برخي محدوديت ها نظير حدود متغيرهاي تصميم و محدوديت هاي خطي را مي توان با تصوير کردن فضاي جستجو به گونه اي که تعداد جواب هاي نشدني آن کمينه باشد اعمال نمود. همچنين مي توان عملگرهاي ترکيب و جهش را طوري طراحي کرد که احتمال زايش جواب هاي نشدني از والدين شدني به حداقل ممکن برسد (ميخائيلويکس و جانيکو ۱۹۹۱ [٥٤]). مي توان ترکيبي از اين روش و روش قبلي را نيز به کار گرفت.

در شرايطي که جواب هاي شدني معلوم نبوده و به سادگي نيز قابل شناسايي نباشند مي توان براي جواب هايي که ناقض محدوديت ها هستند بسته به درجه نقض محدوديت ها جرايمي را در نظر گرفت. اين جريمه ها نوعا" پيش از اينکه مقادير شايستگي محاسبه گردند به تابع هدف اضافه مي شوند (گلدبرگ ۱۹۸۹ [٢٤]). اگرچه اين تکنيک روشي را براي هدايت فرآيند جستجو به سمت جواب هاي شدني ارائه مي نمايد، هنگامي که آنها شناخته شده نيستند مقادير اين جريمه ها بسيار وابسته به مسئله مي باشد. ريچاردسون و همکاران (۱۹۸۹) [٦٦] به منظور استفاده از اين روش راهنمايي هايي ارائه کرده اند.

روش ديگر اعمال محدوديت ها توسط پاول و اسکولنيک (۱۹۹۳) [۶۳] پيشنهاد شده است. در اين رويکرد تابع هدف اوليه به گونه اي تغيير مقياس داده مي شود که در ناحيه شدني مقادير کمتر از واحد

را اختيار نموده و به جواب هاي نشدني جريمه هايي بزرگتر از يک تخصيص دهد. سپس در صورتي که جامعه به درستي مرتب شود، به جواب هاي شدني مقادير شايستگي بزرگتري تخصيص خواهد يافت.

(۴-۱-۵-٤). بهينه سازي چند هدفي با استفاده از الگوريتم هاي ژنتيک

ايده استفاده از الگوريتم هاي ژنتيک در يک مسئله چند معياري در سال ۱۹۶۷ توسط روزنبرگ در قالب استفاده از خواص چندگانه[۱] در جستجوي ژنتيک ارائه شد. با اين وجود نخستين کاربرد عملي اين پيشنهاد در سال ۱۹۸۴ توسط شافر صورت گرفت (گلدبرگ ۱۹۸۹ [۲۴]). شافر (۱۹۸۵) [۷۰] در الگوريتم VEGA[۲] جامعه هر نسل را به زيرجامعه هايي با اندازه مساوي و به تعداد اهداف مسئله تقسيم نمود. در هر يک از اين زيرجامعه ها يک هدف مستقلا" مبناي انتخاب کروموزوم ها قرار مي گيرد. در قدم بعد اعضاي اين زيرجامعه ها از طريق عملگرهاي توليد مثل و ترکيب با يکديگر مخلوط شده و نسل بعد از خود را ايجاد مي نمايند. هرچند اين الگوريتم از نظر اجرايي ساده است ولي به لحاظ اينکه انتخاب کروموزوم ها در هر زيرجامعه بدون توجه به اهداف مطرح در ساير زيرجامعه ها صورت مي گيرد، الگوريتم به سمت نقاط مرزي لبه بهينه پارتو متمايل بوده و نمي تواند به خوبي ساير نقاط واقع بر اين لبه را پيدا کند. گلدبرگ (۱۹۸۹) [۲۴] براي رفع اين مشکل ايده استفاده از سورت نامغلوب[۳] را مطرح کرد. در اين روش اعضاي جامعه بر اساس درجه نامغلوب بودن رتبه بندي مي شوند. ابتدا تمام اعضاي نامغلوب جامعه شناسايي شده و علامت گذاري مي شوند. اين اعضا در صدر فهرست قرار گرفته و رتبه ۱ به آنها داده مي شود. سپس اين اعضا کنار گذاشته شده و اعضاي نامغلوب بعدي شناسايي شده و رتبه ۲ مي گيرند. اين روند همچنان ادامه مي يابد تا تمام جامعه رتبه بندي شوند. در اينجا رتبه هر عضو، ملاک انتخاب خواهد بود.

پس از الگوريتم VEGA الگوريتم هاي ژنتيک ديگري نيز براي مسائل بهينه سازي چند هدفي توسعه يافته است. از جمله اين الگوريتم ها مي توان به: HLGA (هالجا و لين ۱۹۹۲ [۲۸])، FFGA (فونسکا و فلمينگ ۱۹۹۳ [۲۳])، NPGA (هورن و نافپليوتيس ۱۹۹۳ [۳۳])، NSGA (سرينيواس و دب ۱۹۹۴ [۷۷]) و SPEA (زيتزلر و تيل ۱۹۹۸ [۹۳]) اشاره نمود.

---

[۱] Multiple Properties
[۲] Vector Evaluated Genetic Algorithm
[۳] Non-dominated Sorting

(۲-۵-۴) شبکه هاي عصبي[۱]

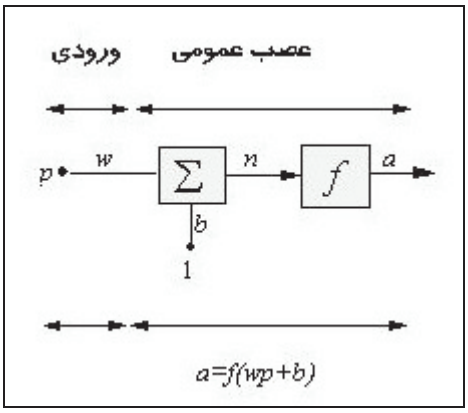
شبکه هاي عصبي شبکه هاي متشکل از اجزاي ساده (معمولا" تطبيق پذير) متصل به هم در يک ساختار سلسله مراتبي هستند که براي تعامل با پديده هاي جهان ، به طريقي مشابه طرز کار سيستمهاي عصبي بيولوژيکي، طراحي شده اند (کوهونن ۱۹۸۸ [۴۶]). در اين قسمت ابتدا ساختار يک شبکه عصبي معرفي شده و پس از آن چگونگي طراحي اين شبکه ها مورد بحث قرار مي گيرند. در ادامه نحوه استفاده از شبکه هاي عصبي در حل مسائل بهينه سازي ارائه شده و امکان استفاده از اين تکنيک در حل مسائل بهينه سازي چند هدفي بررسي مي شود. در خاتمه نحوه استفاده از شبکه هاي عصبي در گروه بندي توصيف مي گردد.

(۱-۲-۵-٤). ساختار يک شبکه عصبي

عصب تک ورودي[۲]: عصب ها مهمترين عنصر شبکه هاي عصبي هستند. شکل (۵-۴) يک عصب تک ورودي را نشان مي دهد. اسکالر ورودي p در اسکالر وزن w ضرب مي شود تا حاصل آن يعني wp به عنوان يکي از ورودي ها به جمع کننده[۳] ارسال شود. ورودي ديگر، ۱، در يک بايياس b ضرب شده و به جمع کننده ارسال مي شود. خروجي جمع کننده n که به آن ورودي خالص[٤] نيز مي گويند توسط يک تابع انتقال[۵] f اسکالر خروجي a را توليد مي کند. خروجي عصب به تابع انتقال انتخاب شده بستگي دارد. بايياس در واقع يک وزن است با اين تفاوت که داراي يک ورودي ثابت ۱ مي باشد. بايياس يک عنصر اصلي يک عصب نبوده و قابل صرف نظر کردن مي باشد. لازم به ذکر است که w و b هر دو پارامترهاي اسکالر قابل تنظيم عصب مي باشند. تابع انتقال نوعا" توسط طراح انتخاب شده و پارامترهاي w و b با استفاده از قواعد يادگيري[٦] به گونه اي تنظيم مي شوند که رابطه ورودي/خروجي منظور خاصي را برآورده سازد. براي اطلاعات بيشتر در مورد توابع انتقال مختلف مي توان به هاگان و همکاران (۱۹۹۶) [۲۷] رجوع نمود.

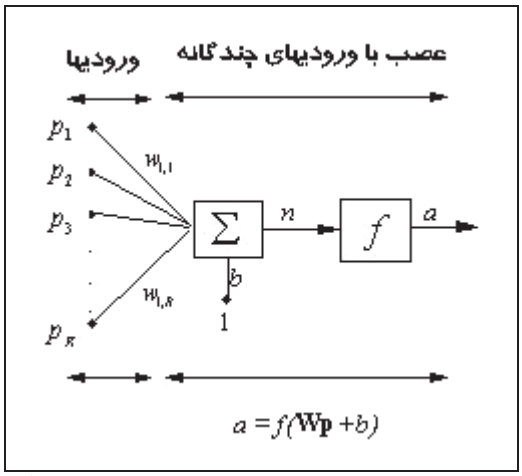
---

[۱] Neural Networks
[۲] Single input neuron
[۳] Summer
[٤] Net input
[۵] Transfer function
[٦] Learning rules



شکل (۵-۴). یک عصب تک ورودي

عصب با ورودي هاي چندگانه[۱]: معمولا" یک عصب بیش از یک ورودي دارد. یک عصب با R ورودي در شکل (٦-۴) نمایش داده شده است. ورودي هاي انفرادي $p_۱, p_۲, ..., p_R$ هر یک در اوزان متناظر $w_{۱,۱}, w_{۱,۲}, ..., w_{۱,R}$ از ماتریس وزن W ضرب شده اند.



شکل (٦-۴). یک عصب با ورودي هاي چندگانه

این عصب داراي یک بایاس b است که در ترکیب با ورودي هاي وزن داده شده، ورودي خالص n را شکل مي دهد:

[۱] Multiple input neuron

$$n = w_{1,1}.p_1 + w_{1,2}.p_2 + \ldots + w_{1,R}.p_R + b \qquad (۴-۷)$$

اين عبارت را مي توان به صورت ماتريسي زير نوشت:

$$n = \mathbf{W}\mathbf{p} + b \qquad (۴-۸)$$

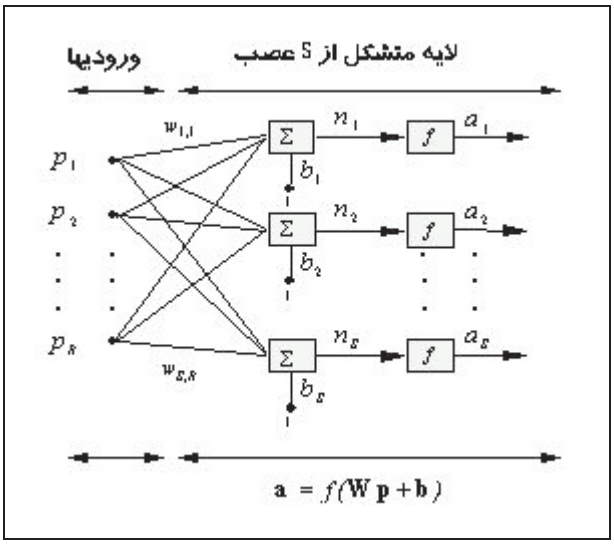
که درآن ماتريس W و بردار p براي عصب تک ورودي تنها يک سطر دارند. حال خروجي عصب را مي توان چنين نوشت:

$$a = f(\mathbf{W}\mathbf{p} + b) \qquad (۴-۹)$$

يک لايه از عصب ها: عموما" يک عصب حتي با چندين ورودي ممکن است کافي نباشد. يک مجموعه از عصب ها که به صورت موازي عمل مي کنند يک لايه[1] ناميده مي شود. يک شبکه تک لايه اي مشتمل بر S عصب در شکل (۴-۷) ارائه شده است. در اين لايه هر يک از R ورودي به تمامي عصب ها متصل شده و ماتريس وزن داراي s سطر مي باشد. اين لايه شامل ماتريس وزن W ، جمع کننده ها ، بردار بايياس b ، توابع انتقال و بردار خروجي a مي باشد. هر يک از عناصر بردار ورودي p از طريق ماتريس وزن W به هر يک از عصب ها متصل شده است. هر عصب داراي يک بايياس b_i ، يک جمع کننده، يک تابع انتقال f و يک خروجي a_i است. خروجي هاي تمام عصب ها مجموعا" بردار خروجي a را شکل مي دهند. معمولا" تعداد ورودي هاي يک لايه با تعداد عصب ها يکسان نيست يعني $R \neq S$. همچنين ضرورتي ندارد که تمام عصب هاي يک لايه داراي يک تابع انتقال باشند .

لايه هاي چندگانه عصب ها: شبکه هاي چند لايه اي قدرتمندتر از شبکه هاي تک لايه اي هستند (هاگان و همکاران ۱۹۹۶ [۲۷]). يک شبکه عصبي سه لايه اي در شکل (۴-۸) نمايش داده شده است.

[1] Layer

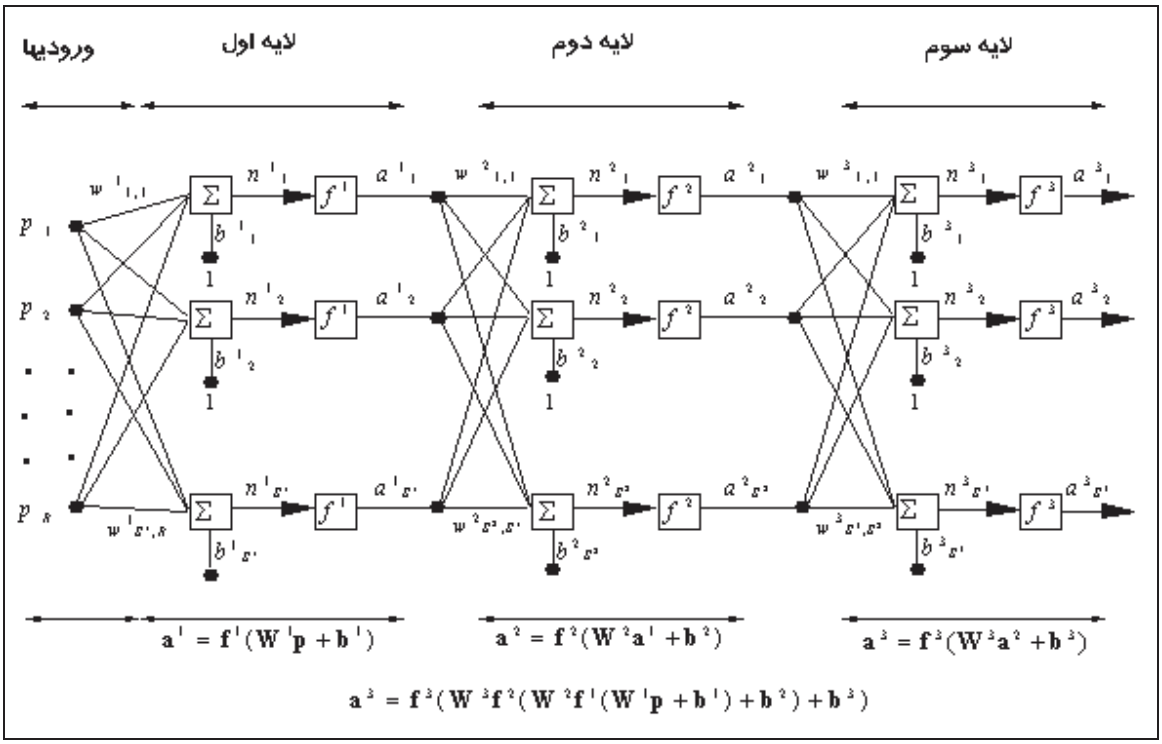


شکل (۷-۴). يک لايه مشتمل بر S عصب

هر لايه داراي ماتريس وزن W، بردار بایاس b ، يک بردار ورودي خالص n و بردار خروجي a خاص خود مي باشد. بالانويس ها براي تمايز لايه ها مختلف استفاده شده اند. به عنوان مثال W^1 ماتريس وزن لايه اول بوده و W^2 و W^3 به ترتيب ماتريسهاي وزن دومين و سومين لايه مي باشند.

همانطور که در شکل نشان داده شده در اين شبکه R ورودي، S^1 عصب در لايه اول،S^2 عصب در لايه دوم و الي آخر وجود دارد. خروجي هاي لايه هاي ۱ و ۲، ورودي هاي لايه هاي ۲ و ۳ هستند. در نتيجه مي توان لايه ۲ را به شکل يک شبکه تک لايه اي با $R = S^1$ ورودي، $S = S^2$ عصب و يک ماتريس $S^1 \times S^2$ وزن $\mathbf{W}^2$ در نظر گرفت. ورودي لايه ۲ بردار $\mathbf{a}^1$ و خروجي آن بردار $\mathbf{a}^2$ مي باشد. لايه اي که خروجي آن خروجي شبکه است، لايه خروجي[1] ناميده مي شود. ساير لايه ها، لايه هاي مخفي خوانده[2] مي شوند. شبکه نشان داده شده در شکل (۸-۴) داراي يک لايه خروجي (لايه ۳) و دو لايه مخفي (لايه هاي ۱ و ۲) مي باشد.

[1] Output Layer
[2] Hidden Layer



شکل (۸-۴). يک شبکه عصبي سه لايه

(۲-۲-۵-٤). طراحي شبکه عصبي

تعداد عصب ها و توابع انتقال: تعداد ورودي ها به شبکه و خروجي ها از شبکه توسط ويژگي هاي خارجي مسئله تعيين مي شوند. در نتيجه اگر چهار متغير خارجي وجود دارند که بايد به عنوان ورودي در نظر گرفته شوند، چهار ورودي به شبکه خواهيم داشت. همچنين اگر هفت خروجي از شبکه نياز داشته باشيم، لايه خروجي داراي هفت عصب خواهد بود. در نهايت، ويژگي مورد نظر در سيگنال خروجي به انتخاب تابع انتقال لايه خروجي کمک مي کند. در نتيجه معماري يک شبکه تک لايه تقريبا" به طور کامل توسط ويژگي هاي مسئله معين مي شود. در مورد شبکه هاي با بيش از دو لايه، مسئله خارجي مستقيما" تعداد عصب ها در لايه هاي مخفي را معين نمي کند. تعيين تعداد بهينه عصب ها در يک لايه مخفي يک زمينه تحقيقاتي فعال مي باشد (هاگان و همکاران ۱۹۹۶ [۲۷]).

معماري شبکه هاي عصبي: معماري هاي مختلفي براي شبکه هاي عصبي ارائه شده است. هاگان و همکاران (۱۹۹۶) [۲۷] طبقه بندي زير را در اين مورد ارائه نموده است: (۱) شبکه هاي تغذيه به جلو[۱]

[۱] Feedforward

(نظير پرسپترون[۱])، (۲) شبکه هاي رقابتي[۲] (مثل شبکه همينگ[۳]) و (۳) شبکه هاي بازگشتي RAM [٤] (نظير شبکه هوپفيلد[۵]).

شبکه هاي تغذيه به جلو براي تشخيص الگو[٦] نيز تقريب تابع بکار برده مي شوند. شبکه هاي رقابتي را مي توان براي دسته بندي ورودي ها در طبقات مختلف آموزش داد. شبکه هاي بازگشتي به عنوان حافظه هاي کمکي همچنين به منظور حل يک گروه از مسائل بهينه سازي مورد استفاده قرار گرفته اند.

يادگيري: قاعده يادگيري (يا الگوريتم آموزش[۷]) رويه اي است براي تغيير وزن ها و بايَاس هاي يک شبکه. هدف قاعده يادگيري، آموزش شبکه به نحوي است که وظايف مشخصي را انجام دهد. هاگان و همکاران (۱۹۹۶) [۲۷] قواعد يادگيري مختلف را در سه گروه بدين ترتيب دسته بندي کرده است: (۱) يادگيري تحت نظارت[۸]، (۲) يادگيري بدون نظارت[۹] و يادگيري تقويت[۱۰] .

در يادگيري تحت نظارت، يادگيري با استفاده از يک مجموعه مثال (مجموعه آموزش) با رفتار شبکه اي مناسب نظير آنچه ذيلا" ارائه شده صورت مي پذيرد:

$$\{\mathbf{p}_1, \mathbf{t}_1\}, \{\mathbf{p}_2, \mathbf{t}_2\}, \ldots, \{\mathbf{p}_Q, \mathbf{t}_Q\}$$

که در آن $\mathbf{p}_Q$ يک ورودي به شبکه و $\mathbf{t}_Q$ خروجي صحيح متناظر با آن (هدف[۱۱]) است. ورودي ها به شبکه اعمال شده و خروجي هاي آن با اين هدف ها مقايسه مي شوند. سپس قاعده يادگيري براي تنظيم وزن ها و بايَاس هاي شبکه بکار گرفته مي شود تا خروجي هاي شبکه را به اين هدف ها نزديکتر سازد.

---

[۱] Perceptron
[۲] Competitive
[۳] Hamming Network
[٤] Recurrent Associate Memory
[۵] Hopefield Network
[٦] Pattern Recognition
[۷] Training algorithm
[۸] Supervised learning
[۹] Unsupervised learning
[۱۰] Reinforcement learning
[۱۱] Target

يادگيري تقويت شبيه يادگيري تحت نظارت است به جز آنکه به جاي اينکه خروجي صحيح هر ورودي داده شده باشد، به الگوريتم تنها يک رتبه داده مي شود. اين رتبه (يا امتياز) معياري براي کارآيي شبکه در برابر دنباله اي از ورودي ها مي باشد. اين نوع يادگيري در حال حاضر کاربرد کمتري نسبت به يادگيري تحت نظارت دارد.

در يادگيري بدون نظارت، وزن ها و بايياس هاي شبکه تنها در پاسخ به ورودي هاي شبکه تغيير مي يابند. در اينجا هيچ خروجي هدفي در اختيار نيست. اغلب اين الگوريتم ها نوعي عمل خوشه بندي[1] را انجام مي دهند. آنها ياد مي گيرند که الگوهاي ورودي را در کلاس هاي محدودي دسته بندي کنند.

شبکه هاي عصبي را مي توان بر حسب نوع داده هاي ورودي (مقادير دودويي يا پيوسته) و روش هاي يادگيري در شش دسته تقسيم بندي کرد. براي مطالعات تکميلي در اين خصوص مي توان به ليپمن (۱۹۸۷) [۵۱] مراجعه نمود.

(۳-۲-۵-٤). کاربرد شبکه هاي عصبي در بهينه سازي

هوپفيلد و تنک (۱۹۸۵) [۳۱] ايده استفاده از شبکه هاي عصبي در حل مسائل بهينه سازي را مطرح نمودند. اين روش بر مبناي تبديل مسئله بهينه سازي به يک تابع انرژي و کمينه سازي آن عمل مي کند.گامهاي زير براي تصوير کردن يک مسئله بهينه سازي در قالب يک شبکه عصبي توسط رامانوجام و سادايايان (۱۹۹۸) [۶۵] پيشنهاد شده است:

۱) طرح نمايشي را انتخاب کنيد که در آن خروجي هاي عصب ها قابل رمزگشايي به صورت يک جواب مسئله باشند.
۲) يک تابع انرژي انتخاب کنيد که مقدار کمينه آن متناظر با بهترين جواب هاي مسئله بهينه سازي باشد.
۳) وزنها و بايياس ها را از تابع انرژي استخراج نمائيد.
٤) مقادير ابتدايي ورودي ها که به طور کامل مقادير خروجي عصب هاي شبکه در حالت پايدار را تعيين مي کند تنظيم نماييد.

[1] Clustering

گامهاي ۱، ۲ و ۳ در تصوير کردن يک مسئله به يک شبکه عصبي بسيار مهم بوده و نقش اساسي در تصميم گيري در اين مورد که آيا مسئله مورد نظر را مي توان به سادگي توسط شبکه عصبي حل نمود يا خير، ايفا مي نمايد.

کاربرد شبکه هاي عصبي در بهينه سازي ترکيبي: در حالت کلي فرض مي شود که يک شبکه عصبي در صورتي که قادر به يافتن يک جواب خوب و نه لزوما" بهينه يک مسئله بهينه سازي ترکيبي[۱] باشد، قادر به حل کردن آن خواهد بود (چيکوچي و آنبهاوئن ۱۹۹۳ [۱۶]). بسياري از مسائل بهينه سازي ترکيبي را مي توان از طريق تعريف توابع انرژي مناسب و تبديل مسئله کمينه سازي به سيستم هاي معادلات ديفرانس[۲] در قالب يک شبکه عصبي تصوير نمود (هوپفيلد و تنک ۱۹۸۵ [۳۱]). تابع انرژي را مي توان به صورت رياضي به صورت زير توصيف کرد:

$$E(\mathbf{x}) = E_c(\mathbf{x}) + \sum_j k_j . E_j(\mathbf{x}) \qquad (۴-۱۰)$$

که در آن $E_c(\mathbf{x})$ تابع هزينه (هدف)، $E_j(\mathbf{x})$ توابع جريمه که بيانگر نقض محدوديت ها هستند، $\mathbf{x} = [x_۱, x_۲, ..., x_n]^T$ بردار طراحي با مقادير گسسته، و $k_j > ۰$ پارامترهاي جريمه که بايد تعادل ميان کمينه سازي تابع هدف و بيشينه سازي تعداد محدوديتهاي ارضا شده را تضمين نمايد مي باشند (چيکوچي و آنبهاوئن ۱۹۹۳ [۱۶]).

در مورد بسياري از مسائل بهينه سازي، تابع انرژي محاسباتي (۱-۵) را مي توان به يک تابع انرژي کوادراتيک به شکل زير تبديل نموده و يا تقريب زد (هوپفيلد و تنک ۱۹۸۵ [۳۱]):

$$E(\mathbf{x}) = -\frac{۱}{۲}\sum_{i=۱}^{n}\sum_{j=۱}^{n} w_{ij} . x_i . x_j - \sum_{i=۱}^{n} \Theta_i . x_i = -\frac{۱}{۲}\mathbf{x}^T \mathbf{W} \mathbf{x} - \Theta^T \mathbf{x} \qquad (۴-۱۱)$$

که در آن:

$$\mathbf{x} = [x_۱, x_۲, ..., x_n]^T \quad , \quad \Theta = [\Theta_۱, \Theta_۲, ..., \Theta_n]^T$$
$$\mathbf{W} = [w_{ij}]_{n \times n} \quad , \quad w_{ii} = ۰ \quad , \quad w_{ij} = w_{ji}$$

[۱] Combinatorial Optimization
[۲] Difference Equation

و x_i مقادير ۱- يا ۱ را مي گيرد $\forall i$. وزن هاي عصبي[۱] و داده هاي ورودي خارجي Θ_i به طور کامل توسط مسئله بهينه سازي در دست حل و يا به طور دقيق تر، توسط تابع انرژي متناظر با آن که بايد کمينه شود تعيين مي گردند.

براي يک تابع انرژي $E(\mathbf{x})$، وزن هاي w_{ij} اتصال عصبي از عصب j به عصب i به صورت زير تعريف مي شوند:

$$w_{ij} = -\frac{\partial^{۲} E(x)}{\partial x_i \partial x_j} \qquad (۴-۱۲)$$

و ورودي Θ_i که از بيرون به عصب i -ام تغذيه مي شود از مشتق اول تابع انرژي بدست مي آيد:

$$\Theta_i = -\left.\frac{\partial E(\mathbf{x})}{\partial x_i}\right|_{\mathbf{x}=۰} \qquad (۴-۱۳)$$

براي جزئيات بيشتر در مورد چگونگي استفاده از شبکه هاي عصبي در حل مسائل بهينه سازي مي توان به هوپفيلد و تنک (۱۹۸۵) [۳۱]، رامانوجام و ساداياپان (۱۹۹۸) [۶۵] و چيکوچي و آنبهاوئن (۱۹۹۳) [۱۶] رجوع نمود.

(۴-۲-۵-٤). بهينه سازي چند هدفي با کمک شبکه هاي عصبي

به منظور بررسي امکان بهينه سازي چند هدفي با استفاده از شبکه هاي عصبي جستجوهاي مختلفي در بانک هاي اطلاعاتي نظير Compendex و EEVL صورت پذيرفت. نتايج بدست آمده حاکي از آن است که تاکنون مورد خاصي از کاربرد شبکه هاي عصبي در پيدا کردن مجموعه جواب هاي نامغلوب يک مسئله بهينه سازي چند هدفي گزارش نشده است. به نظر مي رسد استفاده از شبکه هاي عصبي در بهينه سازي چند هدفي تنها پس از تبديل مسئله چند هدفي به يک مسئله تک هدفي (مثلا" از طريق وزن دهي به اهداف چندگانه و ادغام آنها در قالب يک تابع هدف واحد) ميسر باشد.

[۱] Synaptic Weights

(۶-۴). جمع بندي

در اين فصل مسئله تصميم گيري چند معياري يا MCDM مورد بحث قرار گرفت. خانواده مسائل چند معياري در دو زيرگروه تصميم گيري بر اساس ويژگي هاي چند گانه يا MADM و تصميم گيري چند هدفي يا MODM دسته بندي شدند. مدل چند هدفي طراحي سلولهاي توليدي که در فصل ۳ ارائه شد با گروه مسائل تصميم گيري چند هدفي همخواني دارد و لذا در ادامه بحث روي اين گروه مسائل که به مسائل بهينه سازي چند هدفي يا MOP نيز معروف هستند متمرکز شد. مفاهيم پايه اي در بهينه سازي چند هدفي از جمله غلبه يا چيرگي پارتو، بهينگي پارتو، مجموعه ها و لبه هاي نامغلوب تعريف شدند. هدف روشهاي حل مسائل بهينه سازي چند هدفي پيدا کردن مجموعه جواب هاي نامغلوب است. اين ايده معادل پيدا کردن جواب بهينه در مسائل بهينه سازي تک هدفي است. در ادامه رويکردهاي اساسي در حل مسائل چند هدفي يعني تصميم گيري پيش از جستجو، جستجو پيش از تصميم گيري، و تصميم گيري در حين جستجو معرفي شدند. در رويکرد اول با استفاده از ترجيحات تصميم گير، اهداف مختلف تبديل به يک هدف مي شوند. رويکرد دوم صرف نظر از ترجيحات تصميم گير به جستجوي مجموعه جواب هاي نامغلوب مسئله پرداخته و در انتها آنها را براي انتخاب نهايي به تصميم گير ارائه مي دهد. سومين رويکرد از ترجيحات تصميم گير در حين فرآيند جستجو بهره گرفته و فضاي جواب را به طور پيوسته محدود مي سازد تا جواب مطلوب تصميم گير پيدا شود.

با توجه به وابستگي رويکردهاي مبتني بر ترجيحات تصميم گير به سليقه ها و نظرات شخصي افراد، به منظور حل مدل چند هدفي طراحي سلولهاي توليدي ارائه شده در فصل ۳ از رويکرد جستجو پيش از تصميم گيري استفاده خواهد شد. همچنين به لحاظ محدوديت روشهاي سنتي نظير روش هاي وزن دهي و محدوديت در پيدا کردن مجموعه جواب هاي نامغلوب مسئله، لازم است تکنيک موثري براي پيدا کردن مجموعه جواب هاي نامغلوب اين مسئله جهت ارائه به تصميم گير که همان طراح سيستم توليد ميباشد توسعه يابد. بدين منظور تکنيک هاي الگوريتم ژنتيک و شبکه عصبي به عنوان دو نامزد بالقوه در حل مسئله چند هدفي مورد ارزيابي قرار گرفتند.

در جهت امکان سنجي استفاده از الگوريتم هاي ژنتيک، ابتدا چارچوب يک الگوريتم ژنتيک معرفي شد. سپس کاربرد الگوريتم هاي ژنتيک در بهينه سازي و نحوه اعمال محدوديت هاي مسائل بهينه

سازي در يک الگوريتم ژنتيک مورد بحث قرار گرفت. در انتهاي بخش مربوط به الگوريتم هاي ژنتيک، نحوه استفاده از اين الگوريتم ها در حل مسائل بهينه سازي چند هدفي ارائه شد.

در قسمت بعد، شبکه هاي عصبي مورد بررسي قرار گرفته و در اين راستا ابتدا ساختار يک شبکه عصبي و مسائل مطرح در طراحي يک شبکه عصبي مورد بحث قرار گرفتند. سپس چگونگي استفاده از شبکه هاي عصبي در بهينه سازي ارائه شده و به چگونگي استفاده از اين شبکه ها در بهينه سازي چند هدفي اشاره شد.

بر اساس مطالعات و بررسي هاي انجام شده و با توجه به اين واقعيت که اولا" تصوير کردن يک مسئله بهينه سازي در قالب الگوريتم ژنتيک ساده تر از ساخت يک شبکه عصبي براي حل آن بوده و ثانيا" بهره گيري کامل از مزيت سرعت محاسباتي شبکه هاي عصبي مصنوعي تا حد زيادي وابسته به توسعه پردازشگرهاي موازي[۱] است، استفاده از الگوريتم هاي ژنتيک براي حل مسئله چند هدفي طراحي سلولهاي توليدي به نظر مناسب تر از شبکه عصبي مي رسد.

[۱] Parallel Processors

فصل ۵

توسعه یک الگوریتم ژنتیک براي حل مدل چند هدفي طراحي سلولهاي توليدي

(۱-۵). مقدمه

در اين فصل يک الگوريتم ژنتيک چند هدفي براي حل مسئله طراحي سلول ارائه مي شود. در ابتدا يک نمايش کروموزومي مناسب براي نمايش فضاي جواب مسئله معرفي مي شود. سپس نحوه محاسبه مقادير شايستگي هر جواب در ارتباط با توابع هدف ۴ گانه توصيف مي گردد. بخش اصلي اين فصل به معرفي الگوريتم ژنتيک چند هدفي طراحي شده تحت عنوان XGA مي پردازد. در ادامه کليات نرم افزار کامپيوتري توسعه يافته به زبان Visual C++ براي اجراي الگوريتم XGA شامل ساختمان داده ها و فايل هاي ورودي و خروجي آن معرفي مي شوند. سپس چگونگي تنظيم پارامترهاي الگوريتم با استفاده از ۲ مسئله نمونه توصيف مي شود. و بالاخره در انتها جمع بندي و نتيجه گيري ارائه مي گردد.

(۵-۲). نمايش كروموزومي

براي نمايش فضاي جواب از یک ساختار كروموزومي مشتمل بر n ژن استفاده شده است كه در آن n تعداد متغيرهاي مسئله است. از آنجا كه متغيرهاي تصميم از نوع دودويي هستند، كروموزوم هاي متناظر با هر بردار جواب رشته هايي از ۰ و ۱ خواهند بود. به عنوان مثال در مسئله اي با ۴ متغير تصميم: $X_۱$ ، $X_٤$ ، $Y_{۳,۱}$ و $Y_{۷,۲}$ فضاي جواب با استفاده از یک ساختار كروموزومي متشكل از ۴ ژن به صورت زير نشان داده مي شود:

$X_۱$	$X_٤$	$Y_{۳,۱}$	$Y_{۷,۲}$

تعداد جواب هاي اين مسئله ۱۶=$۲^۴$ و كروموزوم هاي متناظر با آنها به شرح زير مي باشند:

۰۰۰۰ ۰۰۰۱ ۰۰۱۰ ۰۰۱۱
۰۱۰۰ ۰۱۰۱ ۰۱۱۰ ۰۱۱۱
۱۰۰۰ ۱۰۰۱ ۱۰۱۰ ۱۰۱۱
۱۱۰۰ ۱۱۰۱ ۱۱۱۰ ۱۱۱۱

(۵-۳). محاسبه مقادير شايستگي انفرادي

به منظور محاسبه مقادير شايستگي انفرادي هر یک از كروموزوم ها در رابطه با توابع هدف ۴ گانه مدل طراحي سلول هاي توليدي، از رابطه نرمال سازي زير استفاده شده است:

$$F_i = \frac{C_i}{C_i + f_i} \quad , \quad i = ۱,...,٤ \qquad (۵-۱)$$

كه در آن F_i مقدار شايستگي نرماليزه شده، f_i مقدار تابع هدف، و C_i ثابت نرمال سازي هدف i مي باشند. به اين ترتيب مقادير شايستگي مابين صفر و یک واقع شده به گونه اي كه مقدار شايستگي یک نشان دهنده مقدار تابع هدف صفر كه كمترين مقدار ممكن است خواهد بود.

(۵-۴). الگوریتم XGA

در این قسمت عملگرهاي ژنتیک و سایر مفاهیم مورد استفاده در الگوریتم XGA معرفي مي شوند.

(۵-۴-۱). نمادگذاري

نمادها و علائم مورد استفاده در الگوریتم به شرح زیر معرفي مي شوند:

Pop : جامعه[۱]

PS : اندازه جامعه[۲] .

ENS_i : تعداد مورد انتظار انتخاب[۳] جواب i .

MP : استخر امتزاج[۴] .

$CNDF$: لبه نامغلوب جاري[۵].

DFV : مقدار شایستگي مجازي[۶].

NC_i : عدد آشیانه[۷] جواب i .

$d_{i,j}$: فاصله مابین دو جواب (کروموزوم) i و j .

NP : پارامتر آشیانه[۸]. حداکثر فاصله بین دو جواب براي آنکه بتوان بین آنها اشتراک برقرار کرد.

$Sh[d_{i,j}]$: مقدار اشتراک میان جواب هاي i و j .

SFV_i : مقدار شایستگي اشتراکي[۹] جواب i .

$PSelect_i$: احتمال انتخاب جواب i .

ES : مجموعه نخبه[۱۰]

ITP : احتمال انتقال اولیه[۱۱] یک جواب.

TP_i : احتمال انتقال جواب i .

---

[۱] Population
[۲] Population Size
[۳] Expected Number of Selection
[۴] Mating Pool
[۵] Current Non-Dominated Front
[۶] Dummy Fitness Value
[۷] Niche Count
[۸] Niching Parameter
[۹] Shared Fitness Value
[۱۰] Elite Set
[۱۱] Initial Transfer Probability

EN : آشیانه اپسیلني[۱] جواب ها در مجموعه نخبه یا ES .

DF : ضریب کاهنده[۲] احتمال انتقال جواب هاي واقع درآشیانه اپسیلني یک جواب انتخاب شده.

MS : مجموعه امتزاج[۳]

ESS : اندازه مجموعه نخبه[٤]

CR : نرخ ترکیب[٥]

IP : احتمال معکوس سازي[٦]

MR : نرخ جهش[۷]

$SNDF$: تعداد لبه هاي نامغلوب متوالي[۸]

Min_SNDF : حداقل تعداد لبه هاي نامغلوب متوالي[۹]

فلوچارت کلي الگوریتم XGA در شکل (۱-۵) به نمایش در آمده است.

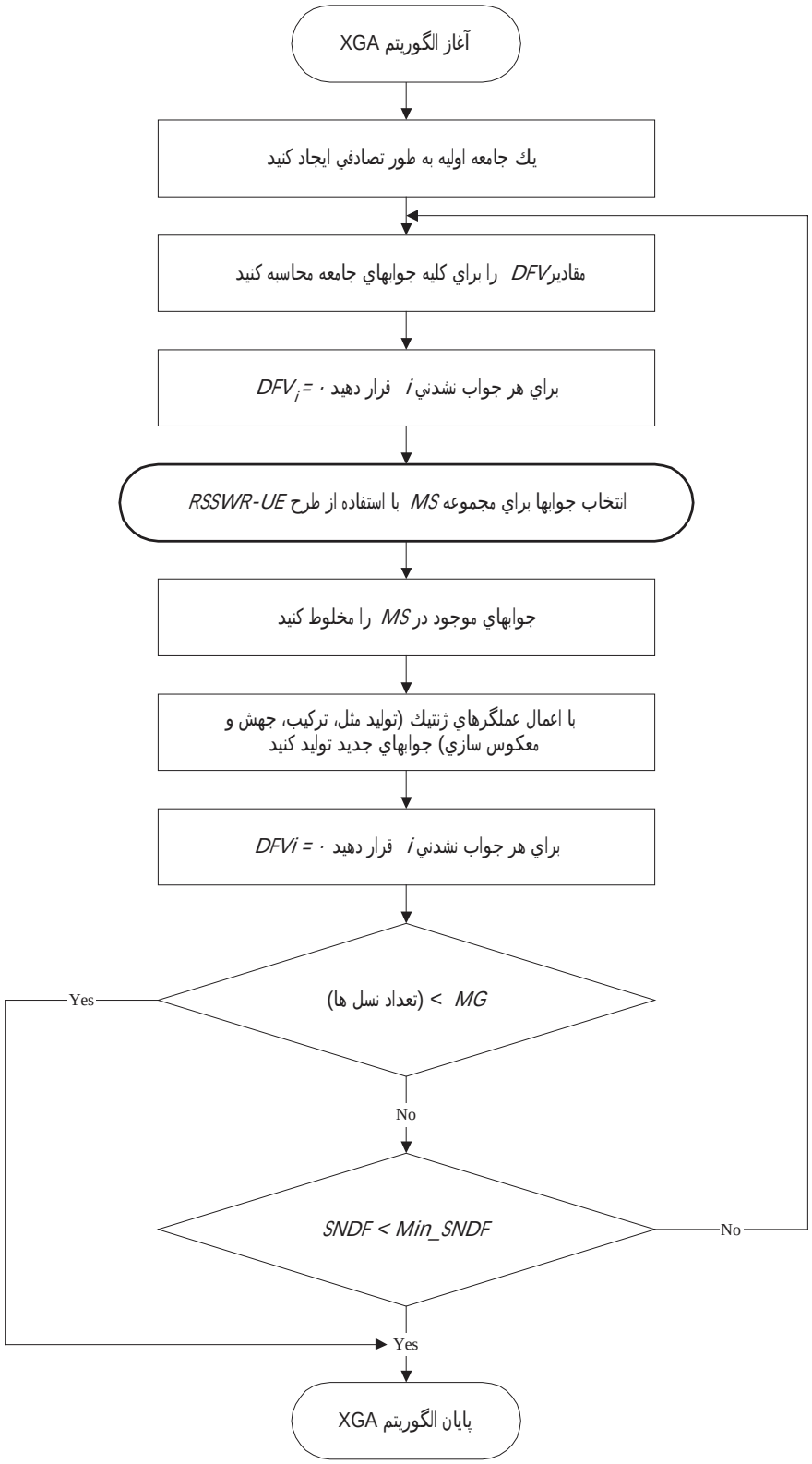
(۲-۴-۵). محاسبه مقدار شایستگي تلفیقي

براي محاسبه مقدار شایستگي تلفیقي کروموزوم ها در یک جامعه با استفاده از مقادیر شایستگي انفرادي متناظر با اهداف ۴ گانه و به منظور هدایت فرآیند جستجو به سمت مجموعه نامغلوب مسئله چند هدفي، از روش پیشنهادي سرینیواس و دب (۱۹۹۴) [۷۷] که بر اساس ایده سورت نامغلوب گلدبرگ (۱۹۸۹) [٢٤] توسعه یافته استفاده شده است. قدم هاي محاسبه مقادیر شایستگي جواب هاي موجود در یک جامعه به شرح زیر است:

۱. لبه نامغلوب جاري $(CNDF)$ متشکل از جوابهاي نامغلوب جامعه را پیدا کرده و به هر یک از اعضاي شدني[۱۰] آن، یک مقدار شایستگي مجازي (DFV) مساوي تخصیص دهید. به جواب هاي نشدني[۱۱] مقدار $DFV = ۰$ تخصیص دهید

---

[۱] Epsilon Niche
[۲] Degrading Factor
[۳] Mating Set
[٤] Elite Set Size
[٥] Crossover Rate
[٦] Inversion Probability
[۷] Mutation Rate
[۸] Successive Non-Dominated Fronts
[۹] Minimum Successive Non-Dominated Fronts
[۱۰] Feasible Solutions
[۱۱] Infeasible Solutions



شكل (۱-۵). فلوچارت كلي الگوريتم XGA

۲. فاصله بين جوابهاي واقع بر لبه نامغلوب جاري $(d_{i,j})$ و مقادير اشتراک[۱] آنها $Sh[d_{i,j}]$ را به صورت زير محاسبه نمائيد:

$$Sh[d_{i,j}] = \begin{cases} ۱ - \dfrac{d_{i,j}}{NP} & \text{if} \quad d_{i,j} \le NP \\ ۰ & \text{Otherwise} \end{cases} \qquad (۵-۲)$$

۳. عدد آشيانه يا NC هر يک از جواب هاي واقع بر لبه نامغلوب جاري را با جمع کردن مقادير اشتراک آن با تمامي جواب هاي واقع بر اين لبه محاسبه نمائيد، يعني:

$$NC_i = \sum_{j \in CNDF} Sh[d_{i,j}] \qquad (۵-۳)$$

٤. مقدار شايستگي اشتراکي[۲] يا SFV هر يک از جواب هاي واقع بر لبه نامغلوب جاري را با تقسيم کردن مقدار شايستگي مجازي بر عدد آشيانه اش محاسبه کنيد، يعني:

$$SFV_i = \frac{DFV}{NC_i} \qquad (۵-۴)$$

۵. اعضاي لبه نامغلوب جاري را کنار گذاشته و قدم هاي ۱ الي ۴ را براي مابقي اعضاي جامعه تکرار کنيد. هنگام تعيين مقدار شايستگي مجازي لبه نامغلوب جاري، عددي کمتر از کوچکترين مقدار شايستگي اشتراکي اعضاي لبه نامغلوب قبلي را تخصيص دهيد.

٦. قدم هاي ۱ الي ۵ را تا آنجا تکرار کنيد که تمام اعضاي جامعه در لبه هاي مختلف دسته بندي شوند.

(۵-۴-۳). پيشگيري از انتخاب جواب هاي نشدني

به منظور اعمال محدوديت هاي مدل و پيشگيري از انتخاب جواب هاي نشدني، مقدار صفر به شايستگي اشتراکي اينگونه جواب ها تخصيص داده مي شود. به اين ترتيب احتمال انتخاب جواب هاي نشدني در مرحله انتخاب از بين مي رود.

---

[۱] Sharing Values
[۲] Shared Fitness Value

(۵-۴-۴). به هنگام سازي مجموعه نخبه

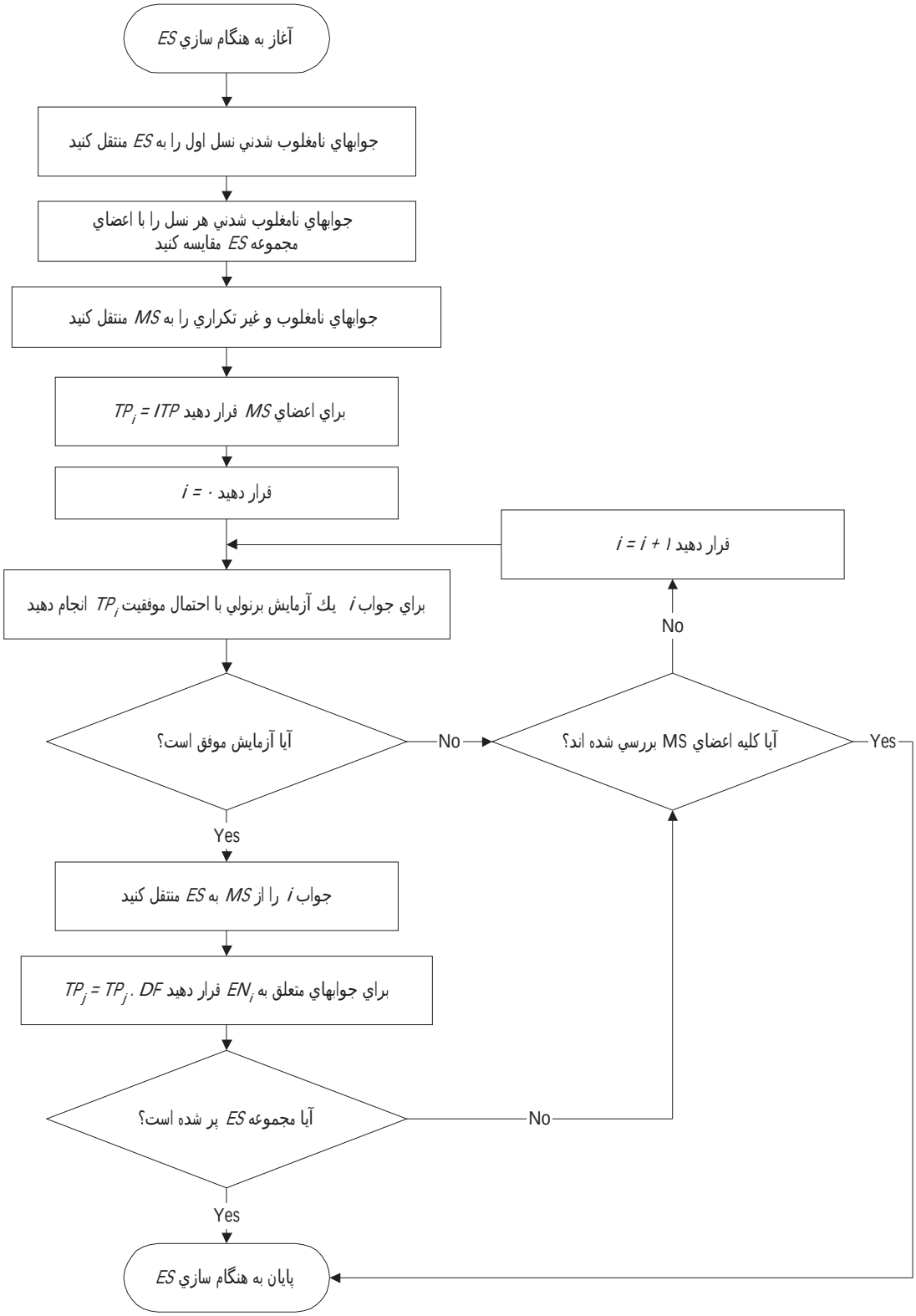
۱. آغاز[۱] : در انتهاي نسل اول تمام جواب هاي نامغلوب شدني را به مجموعه نخبه يا ES منتقل کنيد.
۲. مجموعه جواب هاي نامغلوب شدني پيدا شده در هر نسل را با جواب هاي مجموعه ES مقايسه کرده و پس از کنار گذاشتن جواب هاي مغلوب احتمالي، مابقي جواب ها را به همراه مجموعه جواب هاي موجود در ES به مجموعه امتزاج MS منتقل کنيد.
۳. جواب هاي تکراري را از MS حذف کنيد.
٤. احتمال انتقال از مجموعه MS به مجموعه ES را براي همه جواب هاي متعلق به MS مساوي احتمال انتقال اوليه يا ITP قرار دهيد، يعني: $TP_i = ITP \ , \ \forall i \in MS$.
۵. در مورد هر جواب $i \in MS$ يک آزمايش برنولي با احتمال موفقيت TP_i انجام دهيد. در صورت موفقيت آزمايش:
(۵-۱). جواب i را از MS به ES منتقل کنيد.
(۵-۲). احتمال انتقال کليه جواب هاي واقع در همسايگي اپسيلني جواب i يا EN_i را با ضريب کاهنده DF کاهش دهيد، يعني: $TP_j = DF * TP_j \ , \ \forall j \in EN_i$.
٦. قدم ۵ را تا آنجا تکرار کنيد تا مجموعه ES پر شود، يعني به تعداد اندازه مجموعه نخبه ها يا ESS جواب به آن منتقل شده باشد.

شکل (۵-۲) فرآيند به هنگام سازي مجموعه نخبه ها را در قالب يک فلوچارت نشان مي دهد.

(۵-۴-۵). انتخاب

به منظور انتخاب جواب ها از يک طرح جديد انتخاب بنام $RSSWR-UE$[۲] که ترکيبي از روش $RSSWR$[۳] و يک عملگر جديد نخبه گرايي است استفاده شده است. در اين طرح ابتدا احتمال انتخاب وتعداد مورد انتظار هر يک از جواب ها محاسبه مي شود. سپس مقادير صحيح تعداد مورد انتظار جواب ها برآورده شده و براي انتخاب باقيمانده جواب ها، از يک عملگر جديد نخبه گرا با احتمال مشخص در کنار انتخاب جواب ها بر اساس آزمايشات برنولي روش $RSSWR$ استفاده مي

[۱] Initialization
[۲] Reminder Stochastic Sampling Without Replacement – Using Elitism
[۳] Reminder Stochastic Sampling Without Replacement



شكل (۲-۵). فلوچارت به هنگام سازي مجموعه نخبه ها

شود. عملگر نخبه گرا جواب هايي را از یک مجموعه نخبه[1] با استفاده از یک طرح انتخاب ويژه که در آن از مفهوم آشيانه سازي استفاده شده، انتخاب مي کند. اين مجموعه نخبه در هر نسل به هنگام شده و در عين حال اندازه آن ثابت نگه داشته مي شود. در به هنگام سازي اين مجموعه و حذف جواب هاي اضافي از آن مجددا" از یک طرح انتخاب مبتني بر آشيانه سازي استفاده شده است تا پراکندگي جواب ها در اين مجموعه حفظ شود. قدم هاي طرح انتخاب $RSSWR-UE$ به شرح زير است:

۱. احتمال انتخاب هر یک از جواب ها را با استفاده از مقادير شايستگي اشتراکي به صورت زير محاسبه نمائيد:

$$PSelect_i = \frac{SFV_i}{\sum_{j \in Pop} SFV_j} \qquad (۵-۵)$$

۲. تعداد مورد انتظار انتخاب يا ENS هر یک از جواب ها را به صورت زير محاسبه کنيد:

$$ENS_i = (Pop_Size) * (PSelect_i) \qquad (۵-۶)$$

۳. از هر جواب به تعداد بخش صحيح ENS آن انتخاب کرده و به استخر امتزاج يا MP منتقل کنيد.

٤. براي پر کردن استخر امتزاج MP تا اندازه جامعه يا Pop_Size به شرح زير عمل نمائيد:

(۱-۴). احتمال انتقال يا TP اعضاي ES را مساوي احتمال انتقال اوليه يا ITP قرار دهيد، يعني: $TP_i = ITP \ , \ \forall i \in ES$.

(۲-۴). قرار دهيد $i = ۰$.

(۳-۴). در مورد جواب i یک آزمايش برنولي با احتمال $P_i = ENS_i - [ENS_i]$ انجام دهيد. که در آن $[.]$ نشان دهنده قسمت صحيح مي باشد.

(۱-۳-۴). در صورت موفقيت آزمايش برنولي:

یک آزمايش برنولي با احتمال نخبه گرايي يا PE انجام دهيد.

[1] Elite Set

- در صورت موفقيت آزمايش برنولي، با استفاده از عملگر نخبه گرا يک جواب k از مجموعه نخبه انتخاب کرده و به استخر امتزاج MP منتقل کنيد. احتمالات انتقال جواب هاي موجود در آشيانه اپسيلني يا EN جواب انتخاب شده در مجموعه نخبه را با ضريب کاهنده DF به هنگام کنيد، يعني: $TP_j = DF * TP_j \ , \ \forall \ j \in EN_k$.
- در صورت شکست آزمايش برنولي، جواب i را به عنوان جواب انتخابي به استخر امتزاج انتقال دهيد.

(۲-۳-۴). در صورت شکست آزمايش برنولي، قرار دهيد $i = i + ۱$. اگر استخر امتزاج هنوز پر نشده به قدم (۳-۴) برويد. در غير اين صورت به قدم ۵ برويد.

۵. فرآيند انتخاب را خاتمه دهيد.

شکل (۳-۵) فلوچارت طرح انتخاب $RSSWR\text{-}UE$ را نشان مي دهد.

(۶-۴-۵). مخلوط کردن استخر امتزاج

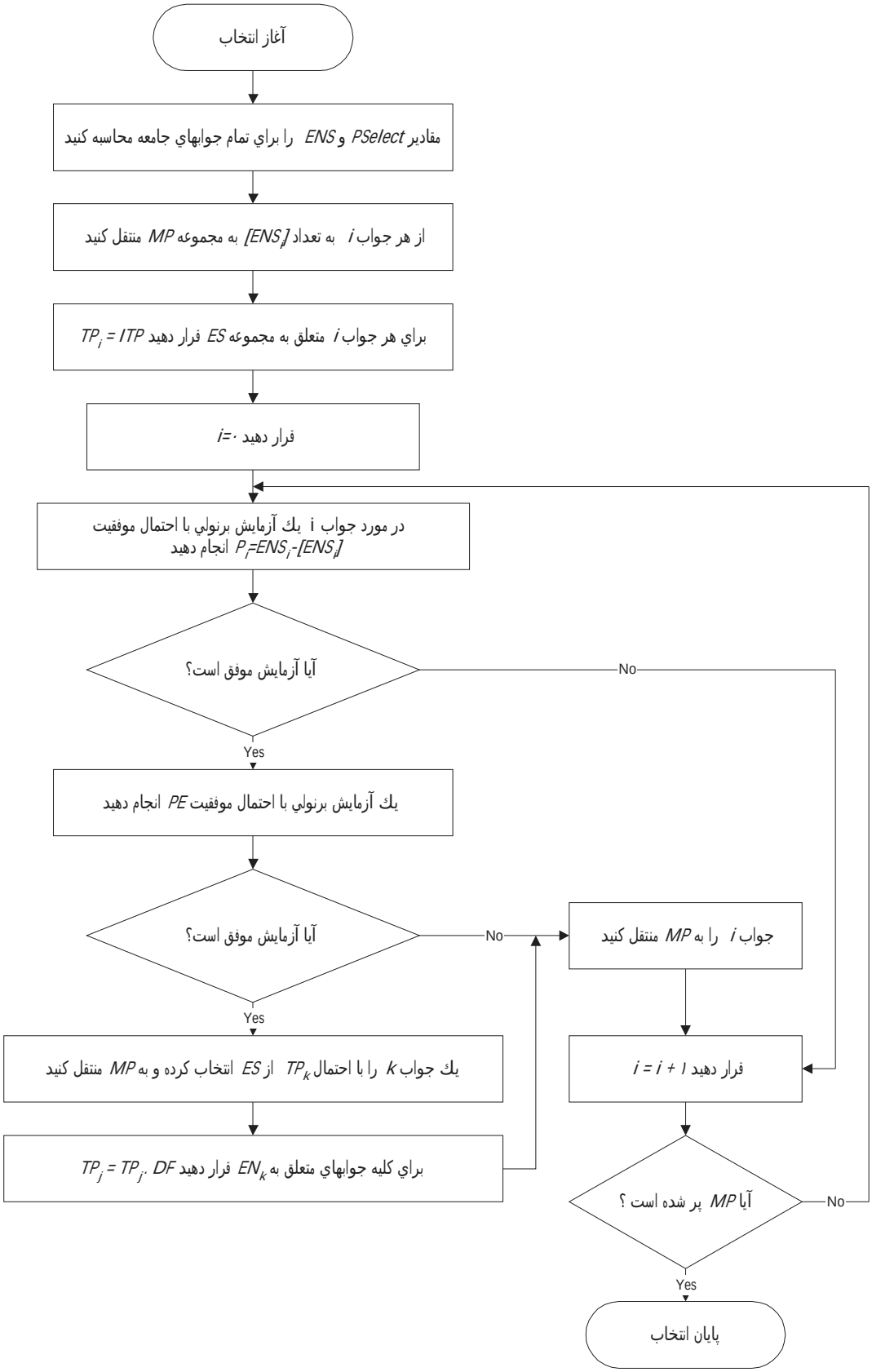
پس از پر شدن استخر امتزاج و به منظور حذف تاثير توالي انتخاب جوابها، استخر امتزاج کاملا مخلوط مي شود به گونه اي که ترتيب قرار گيري جواب ها در آن کاملا" تصادفي باشد.

(۷-۴-۵). ترکيب

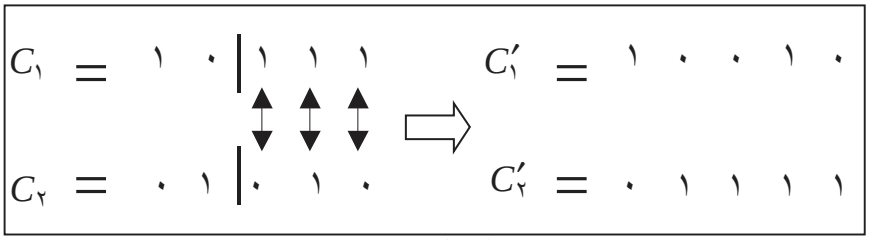
به منظور ترکيب جواب هاي موجود در استخر امتزاج از عملگر ترکيب تک نقطه اي[۱] استفاده مي شود. طرز کار اين عملگر در ترکيب دو جواب فرضي $C_۱$ و $C_۲$ در شکل (٤-۵) نشان داده شده است. مراحل اجراي ترکيب به شرح زير است:

۱. قرار دهيد $i = ۰$.
۲. براي ترکيب جواب هاي i و $(i+۱)$ يک آزمايش برنولي با احتمال موفقيتي معادل نرخ ترکيب يا CR انجام دهيد. در صورت موفقيت به قدم ۳ و در غير اينصورت به قدم ۴ برويد.
۳. موقعيتي را به تصادف در طول کروموزوم هاي i و $(i+۱)$ انتخاب کرده و اين دو کروموزوم را از آن نقطه با هم ترکيب کرده و به نسل بعدي منتقل کنيد.

[۱] Single-Point Crossover



شكل (۳-۵). فلوچارت طرح انتخاب *RSSWR-UE*



شکل (٤-۵). چگونگي ترکيب دو کروموزوم

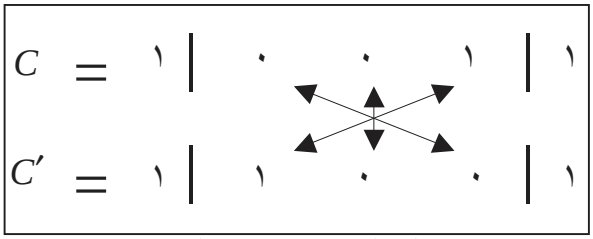
٤. کروموزوم هاي i و $(i+۱)$ را عينا" به نسل بعدي منتقل کنيد.
۵. قرار دهيد $i=i+۲$. در صورتي که $i<(Pop_Size-۱)$ به قدم ۲ برويد. در غير اينصورت توقف کنيد.

(۸-۴-۵). معکوس سازي

پس از ترکيب، عملگر معکوس سازي[۱] به ترتيب زير اجرا مي شود:

۱. قرار دهيد $i=۰$.
۲. براي جواب i يک آزمايش برنولي با احتمال موفقيت معادل احتمال معکوس سازي يا IP انجام دهيد. در صورت موفقيت به قدم ۳ برويد. در غير اينصورت به قدم ۴ برويد.
۳. در طول کروموزوم i دو موقعيت را به تصادف انتخاب کرده و قطعه ژني مابين اين دو نقطه را معکوس کنيد.
٤. قرار دهيد $i=i+۱$.
۵. در صورتي که $i<Pop_Size$ به قدم ۲ برويد. در غير اينصورت توقف کنيد.

نحوه عمل معکوس سازي در شکل (۵-۵) نشان داده شده است.



شکل (۵-۵). چگونگي کارکرد عملگر معکوس سازي

[۱] Inversion

(۹-۴-۵). جهش

عملگر جهش به منظور گسترش فضاي جستجو و کاهش احتمال همگرايي نارس[۱] پس از انجام ترکيب صورت مي پذيرد. بدين منظور روي تک تک ژن هاي کروموزوم هاي حاضر در نسل جديد يک آزمايش برنولي با احتمال موفقيت معادل نرخ جهش يا MR انجام مي پذيرد. در صورت موفقيت اين آزمايش، محتواي ژن عوض مي شود به اين معني که از ۱ به ۰ و بالعکس تغيير مي يابد.

(۱۰-۴-۵). کنار گذاشتن جواب هاي نشدني

از آنجا که احتمال دارد به دنبال اجراي عملگرهاي ترکيب، جهش و معکوس سازي تعدادي جواب نشدني توليد شده باشند، مجددا" شدني بودن جواب ها کنترل شده و به جواب هاي نشدني ضمن علامت گذاري، مقدار شايستگي صفر تخصيص داده مي شود.

(۱۱-۴-۵). معيار توقف

به منظور تصميم گيري در مورد توقف الگوريتم، يک معيار ابتکاري به کار گرفته شده که در آن از تعداد نسل ها و توقف در پيدا کردن جواب هاي غالب جديد به طور همزمان استفاده مي شود. با استفاده از اين معيار، تحقق هر يک از شرايط زير منجر به ختم الگوريتم مي شود:

- شرط ۱: اگر تعداد نسل ها به حداکثر تعداد نسل ها يا MG رسيده باشد.
- شرط ۲: اگر تعداد لبه هاي نامغلوب متوالي يا $SNDF$ در روند زير، به حداقل لبه هاي نامغلوب متوالي يا Min_SNDF برسد:

۱. قرار دهيد $SNDF = ۰$
۲. لبه نامغلوب نسل جاري را با لبه مغلوب ۵ نسل گذشته مقايسه کنيد. اگر کليه جواب هاي موجود در لبه جاري و يا لبه هاي قبلي کماکان نامغلوب باقي ماندند به قدم ۳ برويد. اگر حتي يکي از جواب هاي نامغلوب ۵ نسل اخير در اين مقايسه مغلوب شد به قدم ۴ برويد.
۳. قرار دهيد $SNDF = SNDF + ۱$. نسل بعدي را جايگزين نسل فعلي کرده و به قدم ۲ برويد.
٤. قرار دهيد $SNDF = ۰$. نسل بعدي را جايگزين نسل فعلي کرده و به قدم ۲ برويد.

[۱] Premature Convergence

(۵-۵). نرم افزار کامپيوتري الگوريتم XGA

به منظور اجراي الگوريتم XGA يک نرم افزار کامپيوتري با استفاده از زبان Visual C++ که يک زبان برنامه نويسي شئ گرا[۱] و سطح پايين[۲] است تهيه شده است. اين نرم افزار مشتمل بر ۴۷ تابع و تعداد زيادي متغير است که در ۶ فايل دسته بندي شده اند. متغيرها و برخي توابع مهم نرم افزار در پيوست (۲) ارائه شده اند. در اين قسمت مشخصات کلي و طرز استفاده از اين نرم افزار توصيف مي شوند.

(۱-۵-۵). ساختمان داده ها

متغيرهاي اصلي مسئله در قالب کلاس هاي[۳] زير دسته بندي شده اند:

- کلاس `CPopulation` : اين کلاس حاوي کليه داده هاي مربوط به يک کروموزوم (جواب) مي باشد.
- کلاس `CExceptionalPart` : اين کلاس حاوي داده هاي مربوط به يک قطعه استثنايي مي باشد.
- کلاس `CCell` : اين کلاس حاوي داده هاي مربوط به يک مسئله مي باشد.

اعضاي موجود در کلاس هاي فوق به ترتيب در جداول (۱-۵) تا (۳-۵) معرفي شده اند. ساير متغيرهاي مورد استفاده در نرم افزار به همراه شئ هاي ساخته شده بر اساس کلاس هاي فوق الاشاره در پيوست (۲) آمده اند.

(۲-۵-۵). ورودي و خروجي هاي نرم افزار

داده هاي مربوط به هر مسئله در يک فايل نوشتاري[۴] ذخيره مي شود. فايل هاي ورودي توسط يکي از توابع نرم افزار خوانده شده و مورد استفاده قرار مي گيرد. شکل (۶-۵) محتواي فايل ورودي يک مسئله کوچک با ۴ متغير تصميم مربوط به مثال ارائه شده در فصل ۳ را نشان مي دهد.

---

[۱] Object Oriented
[۲] Low Level
[۳] Class
[۴] Text file

مشخصات اعضاي داده اي		
نام متغير	**نوع متغير**	**توضيح**
value	double	ارزش معادل رشته کروموزومي در مبناي ۱۰
String[]	unsigned char	رشته کروموزومي پس از امتزاج
OldString[]	unsigned char	رشته کروموزومي پيش از امتزاج
Nondominated	bool	نشان دهنده مغلوب يا نامغلوب بودن
IgnoredTemporary	bool	براي کنار گذاشتن موقت جواب از بررسي
DummyFitness	float	شايستگي مجازي
SharedFitness	float	شايستگي اشتراکي
NondominatedFront	int	شماره لبه نامغلوب
TransferProbability	float	احتمال انتقال
SelectedForEliteSet	bool	بيانگر انتخاب شدن يا نشدن براي مجموعه نخبه
SuitableForMatingSet	bool	بيانگر مناسب بودن يا نبودن براي مجموعه امتزاج
SourceAlgorithm	unsigned char	شماره الگوريتم (مورد استفاده در ارزيابي کارآيي الگوريتم)
Feasibility	bool	بيانگر شدني يا نشدني بودن
fitnessF1	float	شايستگي متناظر با تابع هدف f_1
fitnessF2	float	شايستگي متناظر با تابع هدف f_2
fitnessF3	float	شايستگي متناظر با تابع هدف f_3
fitnessF4	float	شايستگي متناظر با تابع هدف f_4
ptoX[]	unsigned char*	آرايه اشاره گرهاي به متغيرهاي X_j
ptoY[][]	unsigned char*	آرايه دو بعدي اشاره گرهاي به متغيرهاي Y_{ik}
OU	float	نسبت بهره گيري کلي
UC[]	float	آرايه نسبت بهره گيري سلول ها
مشخصات اعضاي تابعي		
نام تابع	**نوع تابع**	**توضيح**
CPopulation()	constructor	ايجاد کننده يک شئ از کلاس CPopulation
~CPopulation()	destructor	از بين برنده شئ ايجاد شده

جدول (۱-۵). مشخصات اعضاي کلاس CPopulation

خروجي نرم افزار شامل اطلاعات مربوط به جواب هاي هر نسل، مقادير شايستگي، لبه هاي نامغلوب مختلف، جواب هاي انتخاب شده براي امتزاج و نتيجه حاصل از امتزاج آنها نيز در يک فايل نوشتاري ذخيره مي شوند. شکل (۷-۵) نمونه خروجي نرم افزار مربوط به ۲ نسل از روند حل مسئله ارائه شده در شکل (٦-۵) را نشان مي دهد. در اين مثال اندازه جامعه مساوي ۴ در نظر گرفته شده است.

مشخصات اعضاي داده اي		
نام متغير	**نوع متغير**	**توضيح**
PartIndex	int	انديس قطعه استثنايي
EM_IndexArray[]	int	آرايه انديس هاي ماشين هاي گلوگاهي مربوط به قطعه استثنايي
EM_Size	int	تعداد ماشين هاي گلوگاهي مربوط به قطعه استثنايي
مشخصات اعضاي تابعي		
نام تابع	**نوع تابع**	**توضيح**
CExceptionalPart()	constructor	ايجاد كننده يک شئ از كلاس CExceptionalPart
~CExceptionalPart()	destructor	از بين برنده شئ ايجاد شده

جدول (۲-۵). مشخصات اعضاي كلاس CExceptionalPart

مشخصات اعضاي داده اي		
نام متغير	**نوع متغير**	**توضيح**
BM[]	int	آرايه انديس هاي ماشين هاي گلوگاهي
BM_Size	int	تعداد ماشين هاي گلوگاهي
MC[]	int	آرايه انديس هاي ماشين هاي سلول
MC_Size	int	تعداد ماشين هاي سلول
HF[]	int	آرايه انديس قطعات خانواده سلول
HF_Size	int	تعداد قطعات خانواده اين سلول
GF[]	int	آرايه انديس قطعات نيازمند به ماشينهاي سلول از ساير خانواده ها
GF_Size	int	تعداد قطعات نيازمند به ماشين هاي اين سلول از ساير خانواده ها
EP[]	CExceptionalPart	آرايه قطعات استثنايي سلول
ptoEP[]	CExceptionalPart *	آرايه اشاره گرهاي به قطعات استثنايي
EP_Size	int	تعداد قطعات استثنايي سلول
EP_IndexArray[]	int	آرايه انديس قطعات استثنايي سلول
مشخصات اعضاي تابعي		
نام تابع	**نوع تابع**	**توضيح**
CCell()	constructor	ايجاد كننده يک شئ از كلاس CCell
MembershipCheck ()	bool	كنترل كننده عضويت يک عنصر در يک آرايه
FindRelevantIndexForEP()	int	مشخص كننده انديس مناسب براي قطعه استثنايي
~CCell()	destructor	از بين برنده شئ ايجاد شده

جدول (۳-۵). مشخصات اعضاي كلاس CCell

```
PartTypes: ٦
MachineTypes: ٤
Cells: ٢
MaximumCellSize: ٣
Sub[PartTypes]: ٣ ٣ ١ ٤ ٢ ٣
TempDem[PartTypes]: ١٠٠٠٠ ٧٠٠٠ ٨٠٠٠ ١٠٠٠ ٤٠٠٠ ٢٠٠٠
Mch[MachineTypes]: ١٤٠٠٠ ٦٠٠٠ ١٩٠٠٠ ٧٠٠٠

PM[PartTypes][MachineTypes]:
٠ ٠ ٠ ١٠٠٠٠
٠ ٠ ٠ ٠
٠ ٠ ٠ ٠
٠ ١٠٠٠ ٠ ٠
٠ ٠ ٠ ٠
٠ ٠ ٠ ٠

t[MachineTypes][PartTypes]:
٨ ٠ ٧ ٠ ٠ ١
٧ ٠ ٣ ٢ ٠ ٨
٠ ١٠ ٠ ٦ ٥ ٠
٩ ٣ ٠ ١ ٩ ٠
CM[MachineTypes]: ٢١٠٠٠٠ ٢١٠٠٠٠ ٢١٠٠٠٠ ٢١٠٠٠٠
CS[Cells]: ٢ ٢

DataForCell_١:
HostFamilySize: ٣ HostFamily: ١ ٣ ٦
MachineGroupSize: ٢ MachineCell: ٢ ١
GuestFamilySize: ١ GuestFamily: ٤
NofBottleneckMachines: ١ BottleneckMachines: ٤
NofExceptionalParts: ١
FirstExceptionalPart: ١ NofBottlenecksRequired: ١ BottleneckMachinesRequired: ٤

DataForCell_٢
HostFamilySize: ٣ HostFamily: ٢ ٤ ٥
MachineGroupSize: ٢ MachineCell: ٤ ٣
GuestFamilySize: ١ GuestFamily: ١
NofBottleneckMachines: ١ BottleneckMachines: ٢
NofExceptionalParts: ١
FirstExceptionalPart: ٤ NofBottlenecksRequired: ١ BottleneckMachinesRequired: ٢

Nof-X-variables: ٢ X-Indices: ١ ٤
Nof-Y-variables: ٢ Y-Indices: ٤ , ١
٢ , ٢
```

شکل (٦-۵). فايل ورود داده هاي يک مسئله نمونه

```
Generation: 1
Nondominated Front 1: 0, 1, 2,
Nondominated Front 2: 3,
Decoded values of the Selected Strings:  3,  0,  0,  3,
                                          Fitnesses
                          ---------------------------------------------------------
Str.   Decoded   String   F1        F2        F3        F4        Shared    New String
       Value
0      0         0000     0.4762    1.0000    0.7083    0.9999    0.4690    0011
1      3         0011     1.0000    0.8850    0.6219    1.0000    0.7659    0000
2      4         0100     0.5000    0.9615    0.7029    0.9997    0.5475    0000
3      5         0101     0.5000    0.9091    0.6502    0.9767    0.4643    0011

Successive Nondominated Front: 1

Generation: 2
Nondominated Front 1: 0, 1, 2, 3,
Decoded values of the Selected Strings:  3,  0,  0,  3,
                                          Fitnesses
                          ---------------------------------------------------------
Str.   Decoded   String   F1        F2        F3        F4        Shared    New String
       Value
0      3         0011     1.0000    0.8850    0.6219    1.0000    0.3829    0000
1      0         0000     0.4762    1.0000    0.7083    0.9999    0.3829    0011
2      0         0000     0.4762    1.0000    0.7083    0.9999    0.3829    0011
3      3         0011     1.0000    0.8850    0.6219    1.0000    0.3829    0000
Successive Nondominated Front: 2
```

شکل (۷-۵) نمونه خروجي نرم افزار

(۶-۵). تنظيم پارامترها

از آنجا که الگوريتم هاي ژنتيک ماهيت تصادفي[۱] دارند، استفاده از يک مجموعه پارامتر در مسائل مختلف عموما" جواب هاي متفاوتي توليد مي کند. بر اين اساس تنظيم پارامترهاي الگوريتم ژنتيک براي حل مسئله مشخص از اهميت ويژه اي برخوردار است. پيدا کردن يک مجموعه خوب از پارامترها زمان زيادي را از کامپيوتر صرف مي نمايد (ديويس ۱۹۸۹ [۲۰]).

به منظور تنظيم پارامترهاي الگوريتم XGA از دو مسئله طراحي سلول نمونه که از مقالات انتخاب شده اند استفاده شده است. اولين مسئله از مقاله ونوگوپال و نارندران (۱۹۹۲) [۸٤] انتخاب شده و شامل ۳۰ قطعه و ۱۵ ماشين است که در ۳ دسته گروه بندي شده اند. نحوه اين دسته بندي در شکل (۱) پيوست (۳) نشان داده شده است. در اين مسئله ۶ قطعه استثنايي و ۴ ماشين گلوگاهي وجود دارد.

[۱] Stochastic

فضاي جواب اين مسئله با استفاده از یک ساختار کروموزومي ۱۰ ژني به صورت زير نمايش داده مي شود:

$X_{۱}$	$X_{۳}$	$X_{۱۶}$	$X_{۱۳}$	$X_{۱۵}$	$X_{٤}$	$Y_{۱۱,۱}$	$Y_{۳,۲}$	$Y_{۳,۳}$	$Y_{٤,۳}$

مسئله دوم برگرفته از باربيج (۱۹۶۹) [۹] است و همانطور که در شکل (۲) پيوست (۳) نشان داده شده، مشتمل بر ۳۵ قطعه و ۲۰ ماشين است که در ۵ دسته گروه بندي شده اند. در اين مسئله ۸ قطعه استثنايي و ۷ ماشين گلوگاهي وجود دارد و لذا یک ساختار کروموزومي ۱۵ ژني براي نمايش فضاي جواب مورد نياز است. متغيرهاي تصميم اين مسئله و نحوه قرار گرفتن آنها در ساختار کروموزومي به صورت زير مي باشد:

$X_{۳۱}$	$X_{۱۶}$	$X_{۳۳}$	$X_{۳٤}$	$X_{۳۵}$	$X_{۲۳}$	$X_{۸}$	$X_{۱٤}$	$Y_{۳,۱}$	$Y_{۱۱,۳}$	$Y_{۱۲,۳}$	$Y_{٦,۳}$	$Y_{۱۰,۳}$	$Y_{۲۰,٤}$	$Y_{۵,۵}$

ساير داده هاي مورد نياز مسئله از قبيل هزينه هاي مقاطعه فرعي، تکرار ماشين، زمان هاي پردازش و نظاير آن به طور تصادفي توليد شده اند.

(۱-۶-۵). مقياس هاي کارآيي

به منظور ارزيابي کارآيي الگوريتم با ترکيب پارامترهاي مختلف از سه مقياس کارآيي[۱] زير استفاده شده است:

- $MP_{۱}$. **کيفيت جواب** : نسبت جواب هاي نامغلوب واقعي در لبه نامغلوب آخرين نسل توليد شده توسط الگوريتم. اين نسبت از مقايسه جواب هاي واقع در لبه نامغلوب پيدا شده توسط الگوريتم با لبه نامغلوب واقعي حاصل از شمارش کامل[۲] جواب هاي مسئله محاسبه مي شود.
- $MP_{۲}$. **پراکندگي جواب ها**. تعداد جواب هاي نامغلوب (غير تکراري) پيدا شده توسط الگوريتم.
- $MP_{۳}$. **زمان محاسبات.** زمان گرفته شده از CPU کامپيوتر تا محقق شدن معيار توقف الگوريتم بر حسب ثانيه.

[۱] Measures of Performance
[۲] Total Enumeration

مقياس $MP_۱$ به تنهايي نمي تواند کارآيي يک ترکيب پارامتري را را ارزيابي کند و لذا در کنار آن از دو مقياس ديگر يعني $MP_۲$ و $MP_۳$ نيز استفاده شده است. به عنوان مثال دو ترکيب پارامتري i و j که مقياس هاي آنها به شرح زير است را در نظر بگيريد:

	$MP_۱$	$MP_۲$	$MP_۳$
i	۱,۰۰	۱	۱۰,۲
j	۰,۹۵	۲۰	۱۱,۵

ترکيب i گرچه داراي مقياس کيفيت ۱,۰۰ است ولي تنها ۱ جواب را پيدا کرده است. در مقابل ترکيب j عليرغم دارا بودن مقياس کيفيت ۰,۹۵ جواب هاي بيشتري را پيدا کرده است که اين نشان دهنده شناسايي بخش بزرگتري از لبه نامغلوب واقعي مسئله مي باشد و گزينه هاي بيشتري را در اختيار تصميم گير قرار مي دهد. از نظر زمان محاسباتي نيز تفاوت چنداني مابين دو ترکيب مشاهده نمي شود.

(۲-۶-۵). ارزيابي حساسيت الگوريتم نسبت به پارامترها

به منظور بررسي حساسيت الگوريتم XGA نسبت به پارامترها دو مرحله آزمايش به عمل آمد. در مرحله اول پارامترها به صورت جداگانه مورد بررسي قرار گرفتند. در اين روند در هر بار يک پارامتر تغيير داده شده و با ثابت نگه داشتن مابقي پارامترها، به تعداد ۲۰ بار هر يک از مسائل حل شدند. ميانگين مقياس هاي اين ۲۰ بار به عنوان نتيجه پذيرفته شده است. نمودارهاي چگونگي تغييرات مقياس هاي کارآيي در اين آزمايشات در شکل هاي (۱) الي (۶) در پيوست (۱) ارائه شده است که شکل هاي (۱) الي (۳) مربوط به مسئله ونوگوپال و نارندران (۱۹۹۲) [۸٤] و شکل هاي (۴) الي (۶) مربوط به مسئله باربيج (۱۹۶۹) [۹] مي باشند.

در مرحله دوم و به منظور بررسي تاثير متقابل تغيير پارامترها آزمايشاتي با تغيير همزمان ۲ پارامتر بر روي مسئله باربيج (۱۹۶۹) [۹] انجام گرفت. همانند مرحله قبل هر آزمايش ۲۰ بار تکرار شده و ميانگين مقياس هاي حاصله در اين ۲۰ بار به عنوان نتيجه نهايي در نظر گرفته شده است. نمودارهاي

تغییرات حاصل ضرب دو مقیاس MP_1 و MP_2 که تصویري از تعداد جواب هاي نامغلوب پیدا شده توسط الگوریتم را نشان مي دهد در شکل هاي (۷) الي (۱۱) پیوست (۱) ارائه شده است.
بر اساس نتایج بدست آمده در ارزیابي حساسیت کارآیي الگوریتم نسبت به تغییر پارامترها مي توان نکات زیر را بیان نمود:

۱. **اندازه جامعه** *(PS)* : افزایش اندازه جامعه موجب بهبود قابل ملاحظه همزمان کیفیت و پراکندگي جواب هاي نامغلوب مي شود. در عین حال همانطور که انتظار مي رفت، این افزایش موجب افزایش زمان محاسبات نیز مي گردد.
۲. **نرخ ترکیب** *(CR)* : تغییرات نرخ ترکیب تاثیر چنداني روي کیفیت، پراکندگي و زمان محاسبات ندارد.
۳. **نرخ جهش** *(MR)* : افزایش نرخ جهش موجب کاهش کیفیت مي شود. در عین حال این افزایش تا حدود ۰,۰۳ الي ۰,۰۵ موجب بهبود قابل ملاحظه پراکندگي جواب ها مي شود و از این به بعد نرخ بهبود کند مي شود. همچنین افزایش نرخ جهش موجب کاهش زمان محاسبات مي شود که با توجه به تاثیر افزایش نرخ جهش روي کاهش کیفیت و افزایش پراکندگي مي توان نتیجه گرفت که این افزایش تا حدود ۰,۰۳ الي ۰,۰۵ تاثیر مثبت داشته و جلوي همگرایي نارس را مي گیرد.
٤. **پارامتر آشیانه** *(NP)* : افزایش پارامتر آشیانه باعث افزایش کیفیت جواب و کاهش پراکندگي مي گردد. در عین حال تغییرات این پارامتر تاثیر قابل ملاحظه اي روي زمان محاسبات ندارد. بر این اساس و به منظور برقراري تعادل میان کیفیت و پراکندگي جواب ها، مقدار ۰,۶ براي این پارامتر انتخاب مي شود.
۵. **احتمال معکوس سازي** *(IP)* : افزایش احتمال معکوس سازي موجب کاهش همزمان کیفیت و پراکندگي جواب ها مي شود. بنابراین و علیرغم تاثیر مثبت این احتمال بر روي کاهش زمان محاسبات، احتمال معکوس سازي ۰,۰ مناسبترین مقدار تشخیص داده شد.
٦. **حداقل تعداد لبه هاي نامغلوب متوالي** *(Min_SNDF)* : افزایش حداقل تعداد لبه هاي نامغلوب متوالي تا حد ۱۵ الي ۲۰ موجب کاهش کیفیت و افزایش پراکندگي مي شود از این حد به بعد تاثیر تغییرات روي کیفیت و پراکندگي ناچیز مي باشد. همچنین افزایش تعداد لبه هاي نامغلوب متوالي تاثیر مستقیم روي افزایش زمان محاسبات دارد.

۷. **احتمال نخبه گرايي** *(EP)* : افزايش اين احتمال تاثير مثبت و نسبتا" ملايمي روي كيفيت دارد و در عين حال تاثير نبتا" كمي روي كاهش پراكندگي جواب ها دارد. همچنين مشاهده مي شود كه افزايش احتمال نخبه گرايي زمان محاسبات را افزايش مي دهد.

۸. **احتمال انتقال اوليه** *(ITP)* : افزايش احتمال انتقال اوليه از سطح ۰,۰۵ الي ۰,۱۰ به بالا موجب كاهش همزمان كيفيت و پراكندگي شده و همزمان زمان محاسبات را افزايش مي دهد.

۹. **آشيانه اپسيلني** *(EN)* : تغييرات اندازه آشيانه اپسيلني تاثير قابل توجهي روي كيفيت ندارد. در عين حال افزايش آن تا حد ۰,۲ الي ۰,۴ موجب بهبود نسبي پراكندگي جواب ها و كاهش زمان محاسبات مي گردد.

۱۰. **اندازه مجموعه نخبه ها** *(ESS)* : افزايش اندازه اين مجموعه موجب افزايش نسبي كيفيت و كاهش پراكندگي شده و در عين حال زمان محاسبات را تا حدي كاهش مي دهد.

۱۱. **عامل كاهنده** *(DF)* : افزايش عامل كاهنده تا سطح ۰,۸ موجب بهبود همزمان كيفيت و پراكندگي مي گردد. افزايش اين عامل از حد ۰,۸ به بعد تاثير منفي روي هر دو مقياس ذكر شده دارد. ضمنا" افزايش عامل كاهنده موجب كاهش نسبي زمان محاسبات مي گردد.

۱۲. **تاثير متقابل** *NP* **و** *MR* : كاهش همزمان *MR* از ۰,۱ به پايين و افزايش *NP* تاثير مثبت روي تعداد جواب هاي نامغلوب پيدا شده توسط الگوريتم (حاصل ضرب MP_1 و MP_2) دارد.

۱۳. **تاثير متقابل** *MR* **و** *IP* : كاهش همزمان *MR* و *IP* به صورت هموار موجب بهبود تعداد جواب هاي پيدا شده توسط الگوريتم دارد.

۱٤. **تاثير متقابل** *IP* **و** *EP* : كاهش همزمان *IP* و *EP* تاثير مثبت روي تعداد جواب هاي نامغلوب دارد. در اين روند تاثير *IP* به مراتب بيشتر از *EP* است.

۱۵. **تاثير متقابل** *EP* **و** *EN*: به نظر مي رسد اين دو عامل تاثير متقابلي روي يكديگر نداشته و صرفا" افزايش *EP* موجب افزايش تعداد جواب ها دارد.

۱٦. **تاثير متقابل** *EN* **و** *ITP* : كاهش *ITP* تاثير مثبت روي افزايش تعداد جواب هاي پيدا شده دارد. مجددا" مشاهده مي شود كه *EN* تاثير چنداني در اين رابطه ندارد.

۱۷. **تاثير متقابل** *Min_SNDF* **و** *CR* : به نظر مي رسد كه اين دو عامل تاثير متقابلي روي يكديگر نداشته باشند. به علاوه افزايش *Min_SNDF* تا حد مشخصي موجب افزايش

تعداد جواب هاي پيدا شده توسط الگوريتم مي شود. در اين رابطه، عامل CR تاثير چنداني را نشان نمي دهد.

۱۸. **تاثير متقابل** CR **و** ESS : افزايش همزمان اين دو عامل موجب افزايش تعداد جواب هاي نامغلوب پيدا شده دارد كه در اين رابطه تاثير افزايش ESS قابل ملاحظه و تاثير افزايش CR ناچيز بوده است.

۱۹. **تاثير متقابل** ESS **و** EN : به نظر نمي رسد اين دو عامل تاثير متقابلي روي هم داشته باشند.

بر اساس نتايج حاصل از آزمايشات به عمل آمده بر روي مسائل نمونه، مجموعه مناسبي از پارامترها براي الگوريتم XGA به شرح جدول (۴-۵) انتخاب شده اند. لازم به ذكر است كه با توجه به معيار توقف طراحي شده براي الگوريتم XGA كه عمدتا" بر اساس تعداد لبه هاي نامغلوب متوالي يا Min_SNDF عمل مي كند، حداكثر تعداد نسل ها يا MG به طور تجربي و با استفاده از نتايج ارائه شده توسط زيتزلر (۱۹۹۹) [۹۲] معادل ۲۵۰ در نظر گرفته شده است.

پارامترهاي عمومي الگوريتم	
اندازه جامعه *(PS)*	۱۵۰
نرخ تركيب *(CR)*	۰,۵
نرخ جهش *(MR)*	۰,۰۳
پارامترآشيانه *(NP)*	۰,٦
احتمال معكوس سازي *(IP)*	۰,۰
حداقل تعداد لبه هاي نامغلوب متوالي*(Min_SNDF)*	۱۵
پارامترهاي عملگر نخبه گرايي	
احتمال نخبه گرايي *(EP)*	۱,۰
احتمال انتقال اوليه *(ITP)*	۰,۱
آشيانه اپسيلني *(EN)*	۰,۳
اندازه مجموعه نخبه ها *(ESS)*	۵۰
عامل كاهنده *(DF)*	۰,۸

جدول (۴-۵). پارامترهاي الگوريتم XGA

(۷-۵). جمع بندي

در اين فصل يک الگوريتم ژنتيک بنام XGA براي حل مسئله چند هدفي طراحي سلول ارائه شد. در اين راستا ابتدا نحوه نمايش کروموزومي که پيش نياز اوليه استفاده از الگوريتم ژنتيک در حل يک مسئله است معرفي شد. سپس چگونگي محاسبه مقادير شايستگي جواب ها در ارتباط با اهداف ۴ گانه بيان شد. در ادامه الگوريتم XGA به تفصيل توضيح داده شد. در اين الگوريتم از ايده سورت نامغلوب براي تخصيص مقادير شايستگي تلفيقي جواب ها استفاده شده است. همچنين يک رويه جديد انتخاب که ترکيبي از شيوه انتخاب *RSSWR* و يک طرح ابتکاري نخبه گرايي است مورد استفاده قرار گرفته است. به منظور نخبه گرايي، يک مجموعه از جواب هاي نخبه نگهداري شده و همراه با توليد نسل ها، به هنگام مي شود. براي ثابت نگه داشتن اندازه اين مجموعه، از ايده آشيانه سازي براي انتقال جواب هاي نخبه اي که پراکندگي بيشتري را ايجاد نمايند استفاده شده است. محدوديت هاي مسئله از طريق تخصيص مقدار شايستگي صفر به جواب هاي نشدني در مرحله انتخاب اعمال مي شوند. عملگرهاي ترکيب تک نقطه اي، جهش و معکوس سازي براي امتزاج جواب ها مورد استفاده قرار گرفته اند. پس از امتزاج جواب هاي انتخابي و به منظور حذف جواب هاي نشدني احتمالي مجددا" جواب هاي نشدني کنار گذاشته مي شوند. يک معيار توقف ابتکاري با توجه به تعداد لبه هاي نامغلوب متوالي در نسل ها در کنار حداکثر تعداد نسل ها مورد استفاده قرار گرفته است.

براي اجراي الگوريتم XGA يک نرم افزار کامپيوتري به زبان Visual C++ تهيه شده است. ساختمان داده هاي اين نرم افزار که داده هاي اصلي مدل را در قالب ۳ کلاس دسته بندي مي کند همراه با اعضاي داده اي و تابعي اين کلاس ها مورد بحث قرار گرفت. همچنين فايل هاي ورودي و خروجي اين نرم افزار معرفي شدند.

در انتهاي فصل چگونگي تنظيم پارامترهاي الگوريتم با استناد به ۳ مقياس کارآيي کيفيت جواب، پراکندگي جواب و زمان محاسبات مورد بحث قرار گرفته و نتايج حاصله ارائه شدند.

فصل ۶

ارزيابي الگوريتم XGA

(۱-٦). مقدمه

اين فصل به ارزيابي عملكرد الگوريتم توسعه يافته براي حل مدل ٤ هدفي طراحي سلولهاي توليدي يا XGA مي پردازد. در اين ارزيابي، شاخصهاي: كيفيت جواب (MP_1) ، پراكندگي جواب ها (MP_2) و زمان محاسبه (MP_3) در تعدادي مسئله طراحي سلول با سه الگوريتم ژنتيك چند هدفي مورد مقايسه قرار مي گيرند. اين سه الگوريتم به نامهاي: VEGA ، NPGA و NSGA در ادبيات بهينه سازي چند هدفي با كمك الگوريتم هاي ژنتيك عمدتا" به عنوان الگوريتم هاي مبنا جهت ارزيابي عملكرد ساير الگوريتم ها انتخاب مي شوند.

ساختار مطالب اين فصل بدين شرح است: ابتدا الگوريتم هاي مبنا معرفي شده و قدمهاي اجرايي همراه با مجموعه پارامترهاي آنها توصيف مي شود. سپس سخت افزار و نرم افزار مورد استفاده در ارزيابي معرفي مي شود. روش هاي ارزيابي در بخش بعدي مورد بررسي قرار گرفته و در ادامه، مسائل انتخاب شده براي ارزيابي معرفي مي شوند. بالاخره در انتها نتايج ارزيابي ارائه مي شود.

(٢-٦). معرفي الگوريتم هاي مبنا

در اين بخش قدم هاي اصلي الگوريتم هاي مبنا معرفي مي شوند. نمادهاي زير در توصيف الگوريتم هاي مبنا مورد استفاده قرار گرفته است:

Pop : جامعه[١]

PS : اندازه جامعه[٢] .

ENS_i : تعداد مورد انتظار انتخاب[٣] جواب i .

MP : استخر امتزاج[٤] .

$CNDF$: لبه نامغلوب جاري[٥].

DFV : مقدار شايستگي مجازي[٦].

NC_i : عدد آشيانه[٧] جواب i .

$d_{i,j}$: فاصله مابين دو جواب (كروموزوم) i و j .

NP : پارامتر آشيانه[٨]. حداكثر فاصله بين دو جواب براي آنكه بتوان بين آنها اشتراک برقرار كرد.

$Sh[\,d_{i,j}\,]$: مقدار اشتراک ميان جواب هاي i و j .

SFV_i : مقدار شايستگي اشتراكي[٩] جواب i .

$PSelect_i$: احتمال انتخاب جواب i .

CR : نرخ تركيب[١٠]

MR : نرخ جهش[١١]

---

[١] Population
[٢] Population Size
[٣] Expected Number of Selection
[٤] Mating Pool
[٥] Current Non-Dominated Front
[٦] Dummy Fitness Value
[٧] Niche Count
[٨] Niching Parameter
[٩] Shared Fitness Value
[١٠] Crossover Rate
[١١] Mutation Rate

(٦-٢-١). الگوريتم VEGA

اين الگوريتم توسط شافر (١٩٨٥) [٧٠] توسعه يافته است. انتخاب كروموزوم ها در اين الگوريتم از يك استخر امتزاج يا MP صورت مي گيرد. نحوه شكل گيري اين استخر در نسل هاي مختلف بدين شرح است:

١. جامعه را به N زير جامعه[١] مساوي تقسيم كنيد كه در آن N تعداد توابع هدف است.
٢. در هر زير جامعه، شايستگي جواب ها را بر اساس يكي از توابع هدف محاسبه كنيد.
٣. در هر زير جامعه، جواب هاي شايسته تر را با استفاده از يک طرح انتخاب براي انتقال به استخر امتزاج انتخاب نماييد.
٤. استخر امتزاج را كاملا" مخلوط نماييد.

(٦-٢-٢). الگوريتم NPGA

هورن و نافپليوتيس (١٩٩٣) [٣٣] الگوريتم NPGA را ارائه كردند. انتخاب جواب ها در اين الگوريتم به شرح زير صورت مي پذيرد:

١. در هر تكرار ٢ نامزد از جامعه به طور تصادفي انتخاب نماييد.
٢. يک مجموعه مقايسه t_{dom} (كه از آن بنام تورنمنت نيز ياد مي شود) نيز به طور تصادفي از جامعه انتخاب كنيد.
٣. هر يک از نامزدها را با اعضاي مجموعه انتخاب مقايسه نماييد. يكي از حالت هاي زير رخ مي دهد:

- اگر يكي از نامزدها مغلوب شد ولي ديگري نشد، نامزد اخير را براي توليد مثل انتخاب كنيد.
- اگر هيچيک و يا هر دو نامزد توسط تورنمنت مغلوب شدند، نامزد داراي عدد آشيانه كوچكتر را انتخاب نماييد. عدد آشيانه براي جواب i به صورت زير محاسبه مي شود:

$$NC_i = \sum_{j \in Pop} Sh[\, d_{i,j} \,] \qquad (٦-١)$$

[١] Sub-Population

كه در آن:

$$Sh[d_{i,j}] = \begin{cases} ۱ - \dfrac{d_{i,j}}{NP} & \text{if} \quad d_{i,j} \le NP \\ ۰ & \text{Otherwise} \end{cases} \qquad (٦-٢)$$

٤. قدم هاي ١ الي ٣ را تا آنجا تكرار كنيد كه به اندازه PS جواب براي تشكيل استخر امتزاج انتخاب شده باشد.

(٣-٢-٦). الگوريتم NSGA

اين الگوريتم توسط سرينيواس و دب (١٩٩٤) [٧٧] و بر اساس ايده سورت نامغلوب گلدبرگ (١٩٨٩) [٢٤] توسعه يافت. انتخاب جواب ها براي تشكيل استخر امتزاج يا MP در اين الگوريتم با استناد به مقادير شايستگي اشتراكي يا SFV انجام مي شود. مقادير SFV اعضاي جامعه به ترتيب زير محاسبه مي شود:

١. لبه نامغلوب جاري $(CNDF)$ متشكل از جوابهاي نامغلوب جامعه را پيدا كرده و به هر يک از اعضاي آن، يک مقدار شايستگي مجازي (DFV) مساوي تخصيص دهيد.
٣. فاصله بين جوابهاي واقع بر لبه نامغلوب جاري $(d_{i,j})$ و مقادير اشتراک[١] آنها $Sh[d_{i,j}]$ را با استفاده از رابطه (٢-٦) محاسبه نمائيد.
٤. عدد آشيانه يا NC هر يک از جواب هاي واقع بر لبه نامغلوب جاري را با جمع كردن مقادير اشتراک آن با تمامي جواب هاي واقع بر اين لبه محاسبه نمائيد، يعني:

$$NC_i = \sum_{j \in CNDF} Sh[d_{i,j}] \qquad (٦-٣)$$

٥. مقدار شايستگي اشتراكي[٢] يا SFV هر يک از جواب هاي واقع بر لبه نامغلوب جاري را با تقسيم كردن مقدار شايستگي مجازي بر عدد آشيانه اش محاسبه كنيد، يعني:

[١] Sharing Values
[٢] Shared Fitness Value

$$SFV_i = \frac{DFV}{NC_i} \qquad (٦-۴)$$

٦. اعضاي لبه نامغلوب جاري را كنار گذاشته و قدم هاي ۱ الي ۴ را براي مابقي اعضاي جامعه تكرار كنيد. هنگام تعيين مقدار شايستگي مجازي لبه نامغلوب جاري، عددي كمتر از كوچكترين مقدار شايستگي اشتراكي اعضاي لبه نامغلوب قبلي را تخصيص دهيد.
۷. قدم هاي ۱ الي ۵ را تا آنجا تكرار كنيد كه تمام اعضاي جامعه در لبه هاي مختلف دسته بندي شوند.

(٣-٦). مجموعه پارامترهاي مورد استفاده

در ارزيابي الگوريتم XGA و مقايسه آن با الگوريتم هاي مبنا از مجموعه پارامترهاي تنظيم شده براي اين الگوريتم (جدول (٤-٥) فصل ۵) استفاده شده است. پارامترهاي الگوريتم هاي مبنا همان مقادير مورد استفاده در زيتزلر (۱۹۹۹) [۹۲] به شرح ذيل در نظر گرفته شده است:

تعداد نسل ها	:	۲۵۰
اندازه جامعه	:	۱۰۰
نرخ تركيب	:	۰,۸
نرخ جهش	:	۰,۰۱
پارامتر آشيانه	:	۰,۴۸۸۶۲
اندازه تورنمنت	:	۱۰

پارامتر آشيانه با استفاده از توصيه هاي ارائه شده توسط دب و گلدبرگ (۱۹۸۹) [٢٤] با فرض شكل گيري ۱۰ آشيانه مستقل محاسبه شده است. اندازه تورنمنت مورد استفاده در الگوريتم NPGA نيز به توصيه هورن و نافپليوتيس (۱۹۹۳) [۳۳] معادل ۱۰ درصد اندازه جامعه در نظر گرفته شده است.

(٦-٤). نرم افزار و سخت افزار مورد استفاده

برنامه كامپيوتري هر يک از الگوريتم هاي مبنا نيز همانند الگوريتم XGA با استفاده از زبان Visual C++ تهيه شد. كليه الگوريتم ها تحت سيستم عامل Windows ٢٠٠٠ اجرا شده اند. ويژگي هاي اصلي سخت افزار مورد استفاده نيز به شرح زير بوده است:

واحد پردازشگر مركزي (CPU)	:	Pentium II - Celeron
سرعت پردازش	:	٣٣٣MHZ
حافظه دسترسي تصادفي (RAM)	:	٦٤ MB

(٦-٥). روش هاي ارزيابي

به منظور ارزيابي الگوريتم XGA در مقايسه با الگوريتم هاي مبناي NPGA ، VEGA و NSGA از دو روش استفاده شده است. در روش اول يک مجموعه مرجع براي ارزيابي كيفيت جوابهاي الگوريتم ها مورد استفاده قرار گرفته است. در روش دوم جواب هاي الگوريتم ها در حل يک مسئله به صورت دو به دو با هم مقايسه شده است. جزئيات بيشتر اين دو روش ذيلا" توضيح داده شده است.

(٦-٥-١). روش ١ : استفاده از يک مجموعه مرجع

در اين روش يک مجموعه مرجع از جواب هاي نامغلوب براي هر يک از مسائل مورد استفاده در ارزيابي تشكيل مي شود. اين مجموعه در مسائل كوچک (با حداكثر ١٥ متغير) از طريق شمارش كامل[١] ايجاد شده است. در مسائل بزرگ (داراي ١۶ متغير و بيشتر) كه امكان شمارش كامل با استفاده از سخت افزار كامپيوتري مورد استفاده ميسر نبود، مجموعه مرجع هر مسئله در ٥٠ بار تكرار الگوريتم XGA كه در هر تكرار يک مجموعه تصادفي از پارامترها مورد استفاده قرار گرفته توليد شده است. براي تشكيل اين مجموعه، جواب هاي نامغلوب هر تكرار با مجموعه جواب هاي نامغلوب حاصله تا تكرار قبلي مقايسه شده و پس از حذف جواب هاي مغلوب، جايگزين مجموعه قبلي مي شود.

[١] Total Enumeration

نرخ ترکيب	$CR \in [۰,۱]$
نرخ جهش	$MR \in [۰,۰, ۰,۲]$
پارامتر آشيانه	$NP \in [۰,۲, ۰,۸]$
احتمال نخبه گرايي	$PE \in [۰,۱]$
حداقل تعداد لبه هاي نامغلوب متوالي	$Min_SNF \in [۵,۳۰]$

جدول (۱-٦). دامنه پارامترهاي متغير الگوريتم XGA مورد استفاده در توليد مجموعه هاي مرجع

محدوده هاي مجموعه پارامترهاي مورد استفاده در هر تکرار که به طور تصادفي با استفاده از يک توزيع يکنواخت[۱] توليد شده و مورد استفاده قرار گرفته اند در جدول (۱-٦) ارائه شده است. پس از تشکيل مجموعه هاي مرجع، هر مسئله ۲۰ بار توسط هر الگوريتم حل شده و متوسط مقياس هاي کارآيي اين تکرار ها يعني: کيفيت جواب يا $MP_۱$ ، پراکندگي جواب ها يا $MP_۲$ و زمان محاسبات يا $MP_۳$ مورد استناد قرار گرفته است.

(۲-۵-٦). روش ۲ : مقايسات دو به دو

در اين روش مجموعه جواب هاي نامغلوب توليد شده توسط الگوريتم ها در مسائل مختلف با هم مقايسه مي شوند. در هر مقايسه احتمالا" تعدادي از جواب هاي پيدا شده توسط يک الگوريتم، مغلوب يک يا چند جواب پيدا شده توسط الگوريتم ديگر خواهند بود. در اينجا نسبت جواب هاي نامغلوب باقي مانده هر الگوريتم از کل جواب هاي نامغلوب شناسايي شده توسط آن به عنوان مقياس کيفيت جواب يا $MP_۱$ و تعداد جواب هاي نامغلوب غير تکراري به عنوان مقياس پراکندگي جواب ها يا $MP_۲$ در نظر گرفته شده اند. در اين روش متوسط مقياس هاي حاصل از ۲۰ تکرار در يک مسئله مورد استناد قرار گرفته است.

[۱] Uniform Distribution

تعداد سلول ها	تعداد قطعات	تعداد ماشين ها	تعداد متغيرهاي تصميم	كد مسئله
٣	٣٠	١۵	١٠	VN٩٢
۵	٣۵	٢٠	١۵	Bur٦٩
٧	۴٠	٢۴	١٩	KLA٩٧
٣	١٩	١٢	٣٠	ACGV٩١-P٢
۵	۴٣	١۶	٣۵	Sei٨٩
۴	٣۵	٢٠	۴٣	BC٩١

جدول (٢-٦). مشخصات مسائل مورد استفاده در ارزيابي الگوريتم ها

(٦-٦). مسائل مورد استفاده در ارزيابي

به منظور ارزيابي الگوريتم XGA و الگوريتم هاي مبنا، تعدادي مسئله طراحي سلول هاي توليدي از مقالات انتخاب شده است. اين مسائل به گونه اي انتخاب شده اند كه پوشش مناسبي از مسائل از نظر تعداد متغيرهاي تصميم، تعداد ماشين ها، قطعات و سلول ها را در بر گرفته باشد. مشخصات مسائل مورد استفاده در ارزيابي در جدول (٢-٦) ارائه شده است . ماتريس هاي مربوط به چگونگي دسته بندي هاي اين مسائل در پيوست (٣) آمده است. ساير داده هاي مورد نياز به طور تصادفي توليد شده اند.

(٧-٦). نتايج ارزيابي

همانطور كه قبلا" اشاره شد، هر يك از مسائل انتخابي ٢٠ بار توسط الگوريتم XGA و الگوريتم هاي مبنا حل شده و ميانگين مقياسهاي حاصله در اين تكرارها مبناي مقايسه قرار گرفته است.

(١-٧-٦). نتايج حاصل از روش ١

نتايج حاصل از مقايسه با مجموعه هاي نامغلوب مرجع (روش ارزيابي ١) در جدول (٣-٦) ارائه شده است. همچنين چگونگي تغييرات مقياسهاي سه گانه MP_1 ، MP_2 و MP_3 الگوريتم ها در مسائل مختلف به ترتيب در شكلهاي (١-٦)، (٢-٦) و (٣-٦) ارائه شده است. اين شكلها نشان دهنده برتري

كد مسئله	مقياس هاي كارآيي الگوريتم ها											
	NPGA			VEGA			NSGA			XGA		
	MP_1	MP_2	MP_3	MP_1	MP_2	MP_3	MP_1	MP_2	MP_3	MP_1	MP_2	MP_3
VN٩٢	٠,٥٣٢	١٨,٣	١٧١,٧	٠,٤٦٧	٨,٨	٣١,٢	٠,٧٦٦	٢٢,٨	٣٢,٣	٠,٩٨٣	٣٠,٦	١٨,٩
Bur٦٩	٠,٠٨٧	٢٨,٣	٢٤٧,٢	٠,٠٧١	٨,٥	٣٧,٥	٠,٧٠٠	٥٤,٦	٣٨,٢	٠,٧٨٩	٧٥,٣	٢٢,٨
KLA٩٧	٠,٠٣٢	٣٣,٥	٣٠٣,٨	٠,٠٥١	١٣,٤	٤٤,٨	٠,٢٤٠	٥٠,٩	٤٧,٩	٠,٣٩٦	٧٤,٨	١٧,٧
ACGV٩١ - P٢	٠,٠٤٦	٥٦,١	٥١٤,١	٠,٤٤١	٢٩.٧	٣٣,٣	٠,٢٢٠	٦٧,٩	٤٠,١	٠,٥٩٩	١٠١,٤	٢٢,٥
Sei٨٩	٠,٠٧٣	٤٧,٤	٥٦٣,٠	٠,٤٤٥	٣٥,٥	٤٨,٦	٠,٣٥٩	٦٧,٠	٥٨,٢	٠,٦٩٥	١٠٦,٥	٣٠,٧
BC٩١	٠,٠٢٩	٥٧,٥	٧٠١,١	٠,١٦٩	٣٤,١	٥٣,٦	٠,٢١٦	٦٩,٢	٦٧,٢	٠,٥٩٦	٩٧,١	٢٤,٨

جدول (٣-٦). نتايج مقايسه با مجموعه هاي نامغلوب مرجع

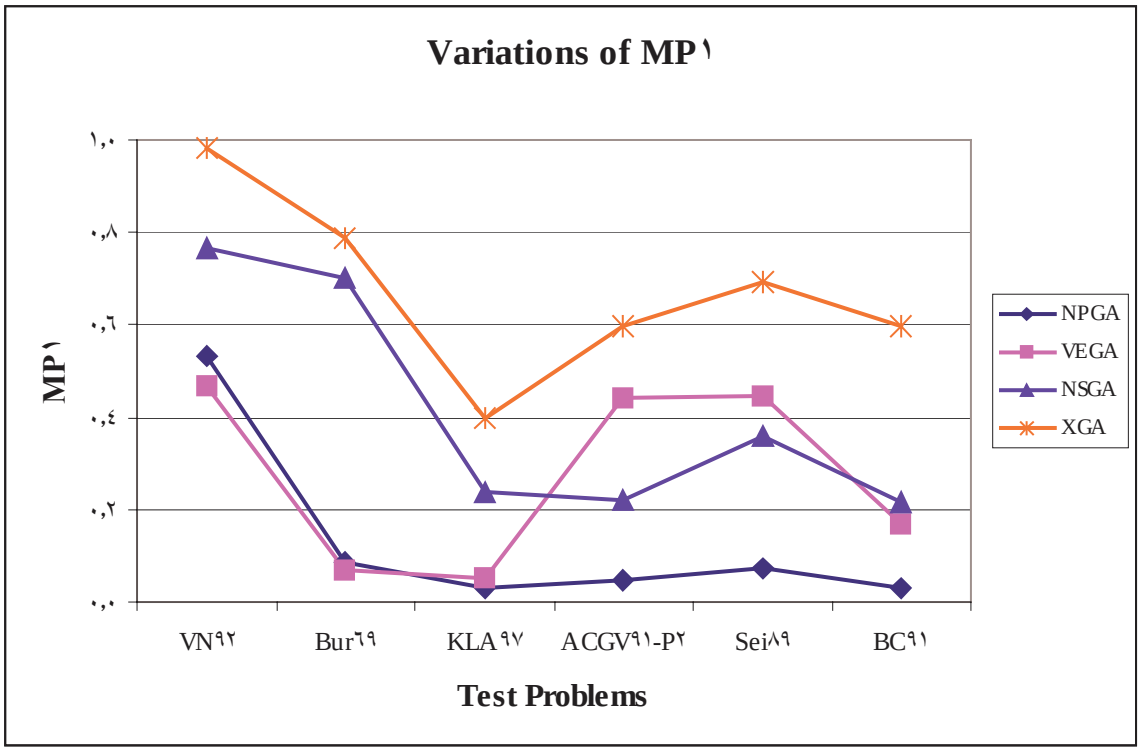
الگوريتم XGA از نظر هر سه مقياس در مسائل آزمايش شده مي باشد. در اينجا نتايج تحليلي مربوط به مقياسهاي سه گانه ارائه مي گردد.

- ✓ **مقياس MP_1 (كيفيت جواب ها)**

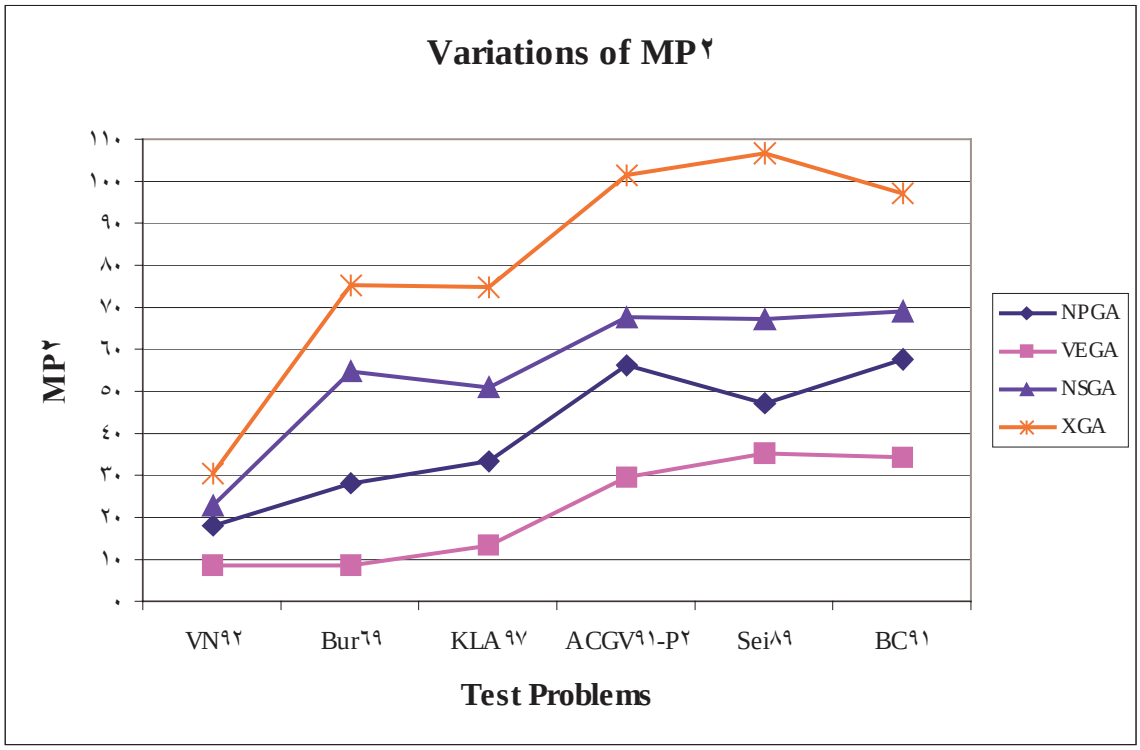
الگوريتم XGA بهترين عملكرد را در تمامي مسائل نشان مي دهد. مقياس MP_1 الگوريتم XGA بين ١٢/٧ تا ١٥٧/٩ در صد بهتر از بهترين نتيجه بدست آمده توسط الگوريتمهاي مبنا بوده است. همچنين نتايج حاصله نشان مي دهد كه الگوريتم NSGA در تمامي مسائل عملكرد بهتري نسبت به الگوريتم NPGA داشته است.

- ✓ **مقياس MP_2 (پراكندگي جواب ها)**

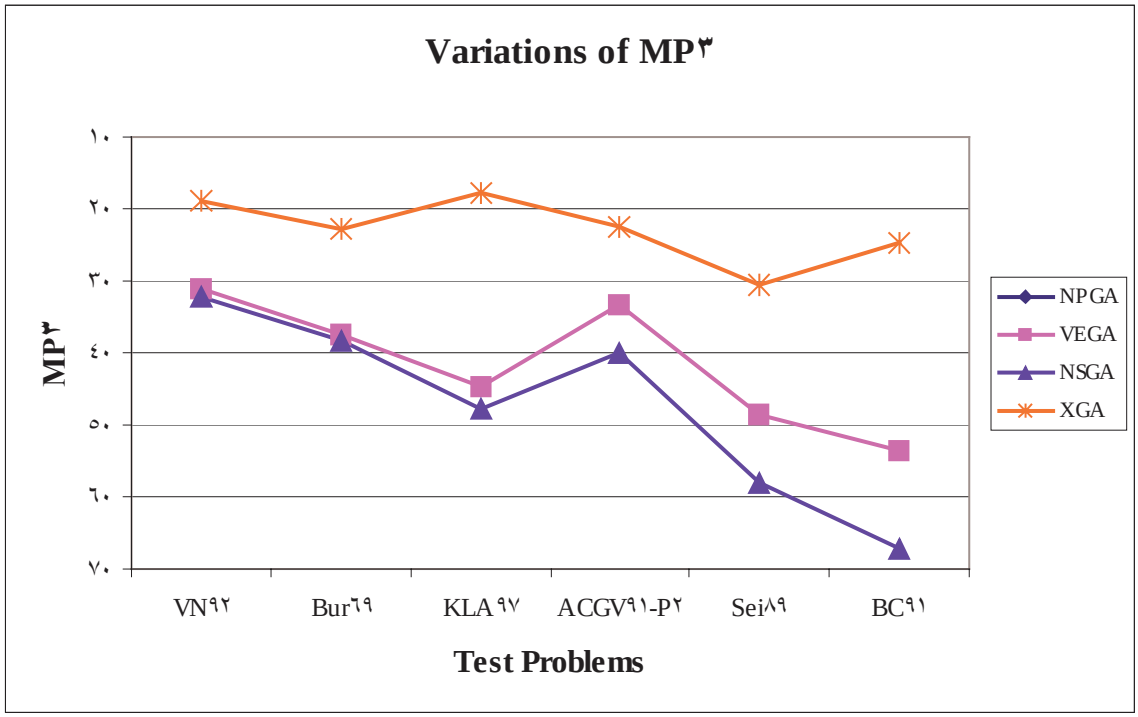
الگوريتم XGA بهترين عملكرد را از نقطه نظر مقياس MP_2 در تمامي مسائل نشان مي دهد به نحوي كه اين مقياس در الگوريتم XGA بين ٣٤/٢ تا ٥٩/٠ در صد بهتر از بهترين نتيجه بدست آمده توسط الگوريتمهاي مبنا بوده است. همچنين در ميان الگوريتمهاي مبنا نيز يك ترتيب منظم مشاهده مي شود به گونه اي كه پس از XGA بهترين نتايج توسط NSGA و پس از آن NPGA و در انتها VEGA حاصل شده است.



شكل (١-٦). چگونگي تغييرات MP_1 در الگوريتم هاي مختلف



شكل (٢-٦). چگونگي تغييرات MP_2 در الگوريتم هاي مختلف



شكل (٣-٦). چگونگي تغييرات MP_3 در الگوريتم هاي مختلف

✓ **مقياس MP_3 (زمان محاسبات)**

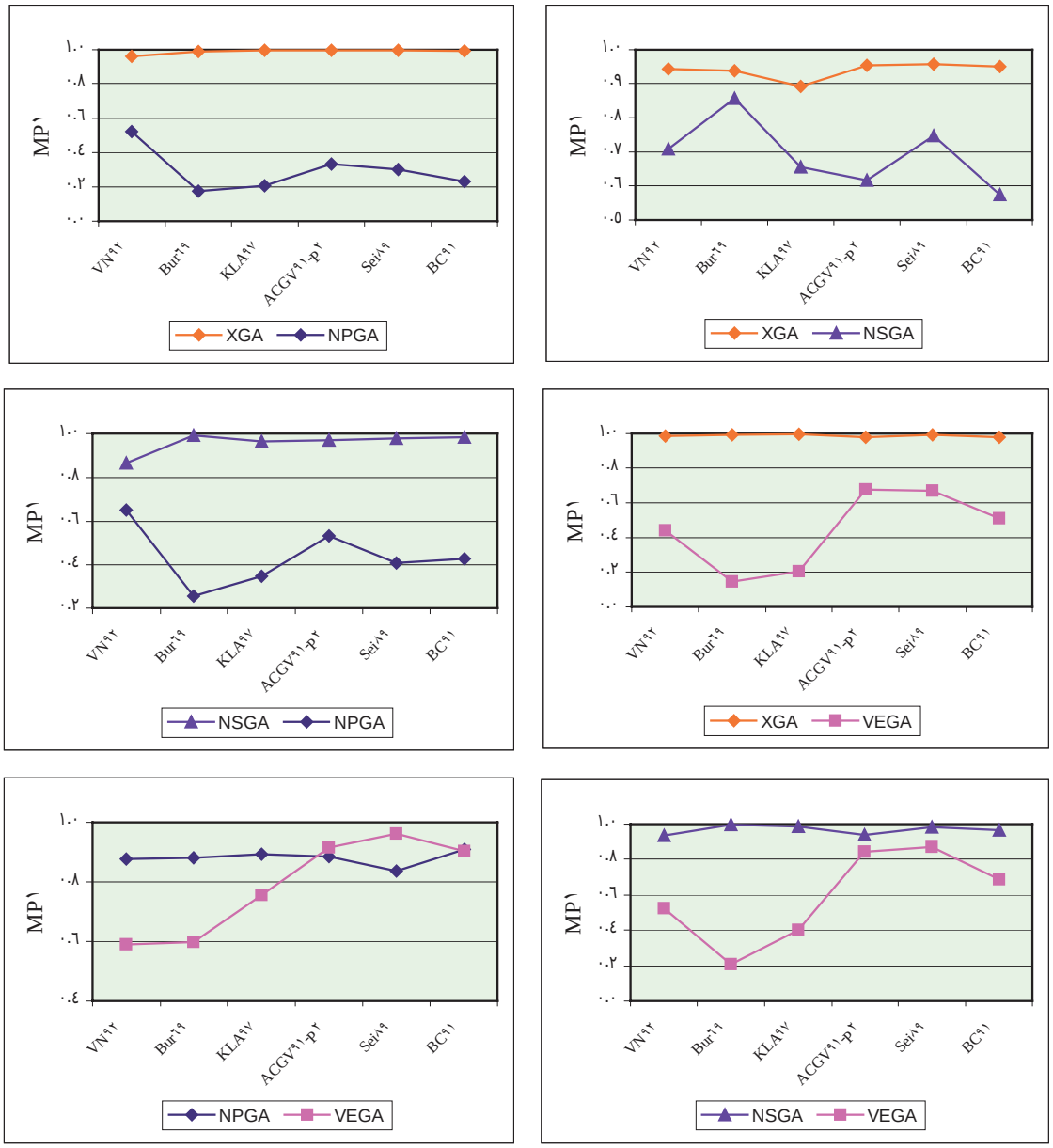
زمان محاسبات الگوريتم XGA كمترين مقدار را در ميان الگوريتم هاي مورد مقايسه داشته است. اين مقياس بين ٣٩/٤ تا ٥٣/٧ درصد بهتر از بهترين نتايج حاصل از الگوريتم هاي مبنا بوده است. در ميان الگوريتم هاي مبنا نيز بهترين زمان در تمام مسائل مربوط به VEGA بوده و پس از آن به ترتيب NSGA و NPGA قرار مي گيرند. زمان هاي صرف شده توسط الگوريتم NPGA اختلاف فاحشي با ساير الگوريتم ها داشته و لذا به منظور وضوح بيشتر، دامنه نقاط مربوط به اين الگوريتم در دامنه مقادير شكل (٣-٦) لحاظ نشده است.

(٢-٧-٦). نتايج حاصل از روش ٢

نتايج حاصل از مقايسات دو به دوي الگوريتم ها (روش ارزيابي ٢) در جدول (٤-٦) ارائه شده است. همچنين چگونگي تغييرات MP_1 در الگوريتم هاي مختلف در شكل (٤-٦) به نمايش در آمده است. اين نتايج نشان مي دهد كه الگوريتم XGA در تمامي مسائل انتخابي مقدار MP_1 بيشتري داشته است. اين برتري در مقايسه با الگوريتم NPGA مابين ٨٣/٧ تا ٤٦٠ درصد، در مقايسه با الگوريتم VEGA حداقل ٤٥/٣ و حداكثر ٥٨٨ درصد ، و بالاخره در مقايسه با الگوريتم NSGA بين ٩/٥ تا

مقايسه	الگوريتم	مقياس	کد مسئله					
			VN٩٢	Bur٦٩	KLA٩٧	ACGV٩١ - P٢	Sei٨٩	BC٩١
XGA - NSGA	XGA	MP_1	٠,٩٤٣	٠,٩٣٨	٠,٨٩٣	٠,٩٥٣	٠,٩٥٨	٠,٩٥٠
		MP_2	٣٠,٦	٧٥,٣	٧٤,٨	١٠٦,٩	١١٠,٧	٩٩,٠
	NSGA	MP_1	٠,٧٠٨	٠,٨٥٧	٠,٦٥٥	٠,٦١٧	٠,٧٤٨	٠,٥٧٥
		MP_2	٢٢,٨	٥٤,٦	٥٠,٩	٧١,٨	٧٢,٣	٧٠,٣
XGA - NPGA	XGA	MP_1	٠,٩٥٩	٠,٩٨٨	٠,٩٩٣	٠,٩٩٢	٠,٩٩٢	٠,٩٩١
		MP_2	٣٠,٦	٧٥,٣	٧٤,٨	١٠٦,٩	١١٠,٧	٩٩,٠
	NPGA	MP_1	٠,٥٢٢	٠,١٧٦	٠,٢٠٥	٠,٣٣٠	٠,٣٠٤	٠,٢٢٩
		MP_2	١٨,٣	٢٨,٣	٣٣,٥	٥٩,١	٤٨,٥	٥٧,٨
XGA - VEGA	XGA	MP_1	٠,٩٨٥	٠,٩٩٢	٠,٩٩٧	٠,٩٧٩	٠,٩٩٢	٠,٩٧٧
		MP_2	٣٠,٦	٧٥,٣	٧٤,٨	١٠٦,٩	١١٠,٧	٩٩,٠
	VEGA	MP_1	٠,٤٤١	٠,١٤٤	٠,٢٠٣	٠,٦٧٤	٠,٦٦٩	٠,٥١٠
		MP_2	٨,٨	٨,٥	١٣,٤	٣٠,٢	٣٥,٥	٣٤,٢
NSGA - NPGA	NSGA	MP_1	٠,٨٦٣	٠,٩٩١	٠,٩٦٣	٠,٩٦٨	٠,٩٧٧	٠,٩٨٣
		MP_2	٢٢,٨	٥٤,٦	٥٠,٩	٧١,٨	٧٢,٣	٧٠,٣
	NPGA	MP_1	٠,٦٥٠	٠,٢٥٨	٠,٣٤٦	٠,٥٣٢	٠,٤٠٩	٠,٤٢٨
		MP_2	١٨,٣	٢٨,٣	٣٣,٥	٥٩,١	٤٨,٥	٥٧,٨
NSGA - VEGA	NSGA	MP_1	٠,٩٣٣	٠,٩٩٦	٠,٩٨٣	٠,٩٤٠	٠,٩٨٣	٠,٩٦٨
		MP_2	٢٢,٨	٥٤,٦	٥٠,٩	٧١,٨	٧٢,٣	٧٠,٣
	VEGA	MP_1	٠,٥٢٣	٠,٢٠٩	٠,٣٩٩	٠,٨٤٤	٠,٨٧٢	٠,٦٨٥
		MP_2	٨,٨	٨,٥	١٣,٤	٣٠,٢	٣٥,٥	٣٤,٢
NPGA - VEGA	NPGA	MP_1	٠,٨٧٦	٠,٨٨٣	٠,٨٩٤	٠,٨٨٥	٠,٨٣٦	٠,٩٠٩
		MP_2	١٨,٣	٢٨,٣	٣٣,٥	٥٩,١	٤٨,٥	٥٧,٨
	VEGA	MP_1	٠,٥٩١	٠,٥٩٨	٠,٧٥٦	٠,٩١٤	٠,٩٦٢	٠,٩٠٣
		MP_2	٨,٨	٨,٥	١٣,٤	٣٠,٢	٣٥,٥	٣٤,٢

جدول (٤-٦). نتايج مقايسات دودويي الگوريتم ها



شكل (۴-٦). چگونگي تغييرات MP۱ در مقايسات دودويي الگوريتم ها

۶۵/۲ درصد بوده است. در ميان سه الگوريتم مبنا نيز NSGA در تمامي مسائل بر دو الگوريتم NPGA و VEGA برتري داشته است. و دست آخر آنكه از بين الگوريتم هاي NPGA و VEGA هيچيک برتري مطلق بر ديگري نداشته اند.

(٨-٦). جمع بندي

در اين فصل عملكرد الگوريتم توسعه يافته براي حل مدل ۴ هدفي طراحي سلولهاي توليدي يا XGA مورد ارزيابي قرار گرفت. در اين ارزيابي، شاخصهاي: كيفيت جواب (MP_1) ، پراكندگي جواب ها (MP_2) و زمان محاسبه (MP_3) در تعدادي مسئله طراحي سلول با سه الگوريتم ژنتيك چند هدفي به نامهاي VEGA ، NPGA و NSGA كه نوعا" در ادبيات بهينه سازي چند هدفي با كمك الگوريتم هاي ژنتيك به عنوان مبنا انتخاب مي شوند مقايسه شد. در اين راستا ابتدا الگوريتم هاي مبنا توصيف شده و سخت افزار و نرم افزار مورد استفاده در ارزيابي معرفي گرديدند. دو روش ارزيابي شامل مقايسه نتايج با يک مجموعه مرجع و مقايسات دودويي الگوريتم ها به عنوان روش هاي ارزيابي ارائه شدند.

نتايج حاصل از روش ارزيابي اول نشان دهنده برتري الگوريتم XGA بر الگوريتم هاي مبنا از نظر هر سه مقياس MP_1 ، MP_2 و MP_3 در تمامي مسائل انتخابي بوده است. همچنين نتايج حاصله نشان مي دهد كه الگوريتم NSGA در تمامي مسائل MP_1 بهتري نسبت به الگوريتم NPGA داشته است. در رابطه با مقياس پراكندگي جواب ها يا MP_2 نيز در ميان الگوريتمهاي مبنا ترتيب منظمي مشاهده مي شود به گونه اي كه پس از XGA بهترين نتايج توسط NSGA و پس از آن NPGA و در انتها VEGA حاصل شده است. و بالاخره از نقطه نظر مقياس زمان محاسبات يا MP_3 در ميان الگوريتم هاي مبنا بهترين زمان در تمام مسائل مربوط به VEGA بوده و پس از آن به ترتيب NSGA و NPGA قرار مي گيرند.

روش ارزيابي دوم كه عمدتا" به ارزيابي مقياس MP_1 مي پردازد نيز مجددا" برتري الگوريتم XGA بر ساير الگوريتم هاي مبنا صحه مي گذارد. در ميان الگوريتم هاي مبنا نيز NSGA در تمامي مسائل بر دو الگوريتم NPGA و VEGA برتري داشته است. و دست آخر آنكه از بين الگوريتم هاي NPGA و VEGA هيچيک برتري مطلق بر ديگري نداشته اند.

فصل ۷

جمع بندي و نتيجه گيري

(۱ - ۷). تعريف مسئله طراحي سلولهاي توليدي

سيستم هاي سنتي توليد را از نظر نحوه آرايش تجهيزات و ماهيت وظايف برنامه ريزي و كنترل توليد به دو دسته سيستم هاي جرياني و سيستم هاي كارگاهي تقسيم بندي شده اند (بلک ۱۹۹۱ [۵]). سيستم جرياني براي توليد انبوه تعداد محدودي از محصولات و سيستم كارگاهي براي توليد محصولات متنوع در تعداد كم مناسب مي باشند. سيستم توليد سلولي سيستمي نسبتا" جديد براي پر كردن محدوده خالي مابين فضاي كاري سيستم هاي سنتي كارگاهي و جرياني است، يعني جايي كه تنوع و نرخ تقاضاي محصولات در حد متوسط مي باشند. سيستم سلولي معمولا" از تغيير سازمان سيستم كارگاهي شكل مي گيرد (بلک ۱۹۹۱ [۵]). از مهمترين مزاياي اقتصادي اين تغيير مي توان به موارد زير اشاره نمود:

- كاهش سهم سرمايه ثابت قطعات از طريق بهره گيري بيشتر از ظرفيت هاي ماشيني
- كاهش سرمايه در گردش به دنبال كاهش حجم موجودي مواد اوليه و قطعات نيم ساخته
- كوتاه شدن دوره برگشت سرمايه به خاطر كاسته شدن از طول زمان ساخت

دستيابي به مزاياي سيستم توليد سلولي درگرو سه تغيير اساسي در زمينه هاي زير است (باربيج ۱۹۷۵ [۹]):

- تغيير نحوه آرايش و سازماندهي ماشين آلات و تشكيل سلولهاي توليدي
- تغيير سيستم سفارش دهي مواد وكنترل موجودي ها
- تغيير سيستم برنامه ريزي توليد

ومرلوف و هير (۱۹۸۷) [۸۷] تغيير اول را جزء مسائل طراحي ساختار سيستم و تغييرات دوم و سوم را در زمره مسائل طراحي ويژگي هاي عملياتي سيستم بر شمرده اند. آنان همچنين انتخاب جامعه قطعات و گروه بندي آنها در قالب خانواده هاي قطعه و انتخاب ماشين ها و فرآيندهاي ساخت و گروه بندي آنها در قالب سلولهاي توليدي را به عنوان دو تصميم اساسي در مرحله طراحي ساختار سيستم توليد سلولي مطرح مي نمايند. شناسايي و تعيين خانواده هاي قطعه و گروه هاي ماشيني در طراحي يک سيستم توليد سلولي به مسئله طراحي سلول هاي توليدي و يا تشكيل سلولهاي توليدي (راجاماني و همكاران ۱۹۹۶ [٦٤]) معروف است.

مسئله طراحي سلولهاي توليدي از جمله زمينه هاي تحقيقاتي در دهه هاي ۸۰ و ۹۰ ميلادي بوده است. در دهه ۸۰ بيشتر مدلها و روشهاي طراحي سلول با در نظر گرفتن صرفا" يک معيار توسعه يافتند. از آغاز دهه ۹۰ تلاش هايي براي مدل سازي و حل اين مسئله در حالت چند معياري صورت گرفته است.

(۲-۷). جمع بندي

در اين رساله ابتدا مسئله طراحي سلولهاي توليدي با تاكيد ويژه بر روشهاي چند معياري مرور شد. به اين منظور ابتدا سيستم توليد سلولي معرفي شده و جايگاه آن در مقايسه با سيستم هاي سنتي كارگاهي و جرياني بر روي نمودار حجم توليد _ تنوع توليد مشخص گرديد. سپس مسئله عمومي طراحي سلولهاي توليدي معرفي شده و مروري بر كارهاي گذشته در حل اين مسئله در حالت تک هدفي ارائه شده و مقالات منتخبي در اين ارتباط معرفي شدند. در ادامه مدلهاي چند معياري طراحي سلولهاي توليدي كه زمينه اصلي اين رساله است با استفاده از يک طرح ابتكاري مقايسه شدند. در اين طرح مقايسه مابين مدلهاي طراحي سلول بر اساس: ورودي ها، معيارها، روش حل، و خروجي هاي مدلها انجام مي شود.

در ادامه ، مسئله چگونگي برخورد با عناصر استثنايي در طراحي يک سيستم توليد سلولي مورد بحث قرار گرفت. تکرار ماشين هاي گلوگاهي و مقاطعه فرعي قطعات استثنايي به عنوان دو سياست پايه اي براي کاهش وابستگي سلولهاي توليدي به يکديگر معرفي شدند. اگرچه کاهش وابستگي سلولها موجب کاهش حرکات بين سلولي به عنوان يکي از معيارهاي مهم در طراحي سيستم توليد سلولي مي شود، بر روي تعدادي از پارامترهاي سيستم از جمله هزينه تبديل سيستم و نسبت بهره گيري ماشين آلات تاثير منفي مي گذارد. براي رفع اين مشکل و ايجاد بستري منطقي براي تصميم گيري در خصوص بهترين استراتژي با توجه به معيارهاي مختلف، يک مدل ۴ هدفي براي پيدا کردن جواب هاي نامغلوب مسئله طراحي سيستم توليد سلولي ارائه شد. اهداف زير در انتخاب طرح يا طرح هاي سلولي مد نظر قرار گرفته است:

- کمينه سازي حرکات بين سلولي.
- کمينه سازي مجموع هزينه هاي تکرار ماشين ها و مقاطعه فرعي قطعات استثنايي.
- کمينه سازي نسبت عدم بهره گيري از ماشين ها در سيستم سلولي که معادل است با بيشينه سازي نسبت بهره گيري از ماشين ها.
- کمينه سازي اختلاف بار کاري سلول هاي مختلف.

بر اساس مطالعات انجام شده روي مقالات ارائه شده در زمينه طراحي سلول هاي توليدي در حالت چند معياري، تاکنون مدلي با مجموعه معيارها و فرض هاي فوق گزارش نشده است. با توجه به وجود تعداد زياد عناصر استثنايي در مسائل واقعي توسعه و استفاده از روش هاي موثر که قادر باشند مجموعه جواب هاي نامغلوب مسئله را در زمان محاسباتي قابل قبولي پيدا نمايند ضروري مي نمايد.

مسئله بهينه سازي چند هدفي و چگونگي استفاده از الگوريتم هاي ژنتيك و شبکه هاي عصبي در حل آن در تحقيق حاضر مورد بررسي قرار گرفت. در اين روند ابتدا خانواده مسائل چند معياري در دو زيرگروه تصميم گيري بر اساس ويژگي هاي چند گانه يا MADM و تصميم گيري چند هدفي يا MODM دسته بندي شدند. مدل چند هدفي طراحي سلولهاي توليدي با گروه مسائل تصميم گيري چند هدفي همخواني دارد و لذا در ادامه بحث روي اين گروه مسائل که به مسائل بهينه سازي چند هدفي يا MOP نيز معروف هستند متمرکز شد. مفاهيم پايه اي در بهينه سازي چند هدفي از جمله غلبه يا چيرگي پارتو، بهينگي پارتو، مجموعه ها و لبه هاي نامغلوب تعريف شدند. هدف روشهاي حل

مسائل بهينه سازي چند هدفي پيدا كردن مجموعه جواب هاي نامغلوب است. اين ايده معادل پيدا كردن جواب بهينه در مسائل بهينه سازي تک هدفي است. در ادامه رويكردهاي اساسي در حل مسائل چند هدفي يعني تصميم گيري پيش از جستجو، جستجو پيش از تصميم گيري، و تصميم گيري در حين جستجو معرفي شدند. در رويكرد اول با استفاده از ترجيحات تصميم گير، اهداف مختلف تبديل به يک هدف مي شوند. رويكرد دوم صرف نظر از ترجيحات تصميم گير به جستجوي مجموعه جواب هاي نامغلوب مسئله پرداخته و در انتها آنها را براي انتخاب نهايي به تصميم گير ارائه مي دهد. سومين رويكرد از ترجيحات تصميم گير در حين فرآيند جستجو بهره گرفته و فضاي جواب را به طور پيوسته محدود مي سازد تا جواب مطلوب تصميم گير پيدا شود. در انتها سه روش سنتي براي حل مسائل بهينه سازي چند هدفي از طريق تبديل آنها به يک مسئله تک هدفي يعني روشهاي: وزن دهي، محدوديت و برنامه ريزي آرماني به اجمال معرفي شده و محدوديت هاي اين قبيل روشها مورد بحث قرار گرفت. با توجه به وابستگي رويكردهاي مبتني بر ترجيحات تصميم گير به سليقه ها و نظرات شخصي افراد، به منظور حل مدل چند هدفي طراحي سلولهاي توليدي ارائه شده در اين رساله از رويكرد جستجو پيش از تصميم گيري استفاده شده است. همچنين به لحاظ محدوديت روشهاي سنتي نظير روش هاي وزن دهي و محدوديت در پيدا كردن مجموعه جواب هاي نامغلوب مسئله، توسعه تكنيک موثري براي پيدا كردن مجموعه جواب هاي نامغلوب اين مسئله جهت ارائه به تصميم گير كه همان طراح سيستم توليد ميباشد ضروري نمود.

در ادامه، تكنيک الگوريتم ژنتيک در حل مسئله چند هدفي طراحي سلول مورد بحث در اين رساله بررسي شد. در جهت امكان سنجي استفاده از الگوريتم هاي ژنتيک، ابتدا چارچوب يک الگوريتم ژنتيک معرفي شده و پس از آن كاربرد الگوريتم هاي ژنتيک در بهينه سازي و نحوه اعمال محدوديت هاي مسائل بهينه سازي در يک الگوريتم ژنتيک مورد بحث قرار گرفت. در انتها ، نحوه استفاده از اين الگوريتم ها در حل مسائل بهينه سازي چند هدفي مورد بحث قرار گرفت. بر اساس مطالعات و بررسي هاي انجام شده ، الگوريتم ژنتيک به عنوان روشي مناسب براي حل مدل چند هدفي توسعه داده شده در اين تحقيق انتخاب گرديد و اين روش جهت مسئله مذكور توسعه داده شد.

در اين خصوص يک الگوريتم ژنتيک بنام XGA براي حل مسئله چند هدفي طراحي سلول ارائه شد. در اين راستا ابتدا نحوه نمايش كروموزومي كه پيش نياز اوليه استفاده از الگوريتم ژنتيک در حل يک مسئله است معرفي شد. سپس چگونگي محاسبه مقادير شايستگي جواب ها در ارتباط با اهداف ۴ گانه

بيان شد. در اين الگوريتم از ايده سورت نامغلوب براي تخصيص مقادير شايستگي تلفيقي جواب ها استفاده شده است. همچنين يک رويه جديد انتخاب که ترکيبي از شيوه انتخاب *RSSWR* و يک طرح ابتکاري نخبه گرايي است مورد استفاده قرار گرفته است. به منظور نخبه گرايي، يک مجموعه از جواب هاي نخبه نگهداري شده و همراه با توليد نسل ها، به هنگام مي شود. براي ثابت نگه داشتن اندازه اين مجموعه، از ايده آشيانه سازي براي انتقال جواب هاي نخبه اي که پراکندگي بيشتري را ايجاد نمايند استفاده شده است.

محدوديت هاي مسئله از طريق تخصيص مقدار شايستگي صفر به جواب هاي نشدني در مرحله انتخاب اعمال مي شوند. عملگرهاي ترکيب تک نقطه اي، جهش و معکوس سازي براي امتزاج جواب ها مورد استفاده قرار گرفته اند. پس از امتزاج جواب هاي انتخابي و به منظور حذف جواب هاي نشدني احتمالي مجددا" جواب هاي نشدني کنار گذاشته مي شوند. يک معيار توقف ابتکاري با توجه به تعداد لبه هاي نامغلوب متوالي در نسل ها در کنار حداکثر تعداد نسل ها مورد استفاده قرار گرفته است.

براي اجراي الگوريتم XGA يک نرم افزار کامپيوتري به زبان Visual C++ تهيه شد. ساختمان داده هاي اين نرم افزار که داده هاي اصلي مدل را در قالب ۳ کلاس دسته بندي مي کند همراه با اعضاي داده اي و تابعي اين کلاس ها مورد بحث قرار گرفت. همچنين فايل هاي ورودي و خروجي اين نرم افزار معرفي شدند.

در ادامه چگونگي تنظيم پارامترهاي الگوريتم با استناد به ۳ مقياس کارآيي کيفيت جواب، پراکندگي جواب و زمان محاسبات مورد بحث قرار گرفته و نتايج حاصله ارائه شدند.

در خاتمه رساله عملکرد الگوريتم توسعه يافته براي حل مدل ۴ هدفي طراحي سلولهاي توليدي يا XGA مورد ارزيابي قرار گرفت. در اين ارزيابي، شاخصهاي: کيفيت جواب (MP_1) ، پراکندگي جواب ها (MP_2) و زمان محاسبه (MP_3) در تعدادي مسئله طراحي سلول با سه الگوريتم ژنتيك چند هدفي به نامهاي VEGA ، NPGA و NSGA که نوعا" در ادبيات بهينه سازي چند هدفي با کمك الگوريتم هاي ژنتيك به عنوان مبنا انتخاب مي شوند مقايسه شد. در اين راستا ابتدا الگوريتم هاي مبنا توصيف شده و سخت افزار و نرم افزار مورد استفاده در ارزيابي معرفي گرديدند. دو روش ارزيابي

شامل مقايسه نتايج با يک مجموعه مرجع و مقايسات دودويي الگوريتم ها به عنوان روش هاي ارزيابي ارائه شدند.

نتايج حاصل از روش ارزيابي اول نشان دهنده برتري الگوريتم XGA بر الگوريتم هاي مبنا از نظر هر سه مقياس MP_1 ، MP_2 و MP_3 در تمامي مسائل انتخابي بوده است. مقياس MP_1 الگوريتم XGA بين ۱۲/۷ تا ۱۵۷/۹ در صد بهتر از بهترين نتيجه بدست آمده توسط الگوريتمهاي مبنا بوده است. همچنين مقياسMP_2 الگوريتم XGA بهترين عملکرد را در تمامي مسائل نشان مي دهد به نحوي که اين مقياس در الگوريتم XGA بين ۳٤/۲ تا ۵۹/۰ در صد بهتر از بهترين نتيجه بدست آمده توسط الگوريتمهاي مبنا بوده است. بالاخره از نظر زمان محاسبات يا مقياس MP_3 الگوريتم XGA کمترين مقدار را در ميان الگوريتم هاي مورد مقايسه داشته است به نحوي که اين مقياس بين ۳۹/٤ تا ۵۳/۷ درصد بهتر از بهترين نتايج حاصل از الگوريتم هاي مبنا بوده است. همچنين نتايج حاصله نشان مي دهد که الگوريتم NSGA در تمامي مسائل MP_1 بهتري نسبت به الگوريتم NPGA داشته است.

در رابطه با مقياس پراکندگي جواب ها يا MP_2 نيز در ميان الگوريتمهاي مبنا ترتيب منظمي مشاهده مي شود به گونه اي که پس از XGA بهترين نتايج توسط NSGA و پس از آن NPGA و در انتها VEGA حاصل شده است.

از نقطه نظر مقياس زمان محاسبات يا MP_3 نيز در ميان الگوريتم هاي مبنا بهترين زمان در تمام مسائل مربوط به VEGA بوده و پس از آن به ترتيب NSGA و NPGA قرار مي گيرند.

نتايج حاصل از مقايسات دو به دوي الگوريتم ها (روش ارزيابي ۲) نشان مي دهد که الگوريتم XGA در تمامي مسائل انتخابي مقدار MP_1 بيشتري داشته است. اين برتري در مقايسه با الگوريتم NPGA مابين ۸۳/۷ تا ۴۶۰ درصد، در مقايسه با الگوريتم VEGA حداقل ۴۵/۳ و حداکثر ۵۸۸ درصد ، و بالاخره در مقايسه با الگوريتم NSGA بين ۹/۵ تا ۶۵/۲ درصد بوده است.

در ميان سه الگوريتم مبنا نيز NSGA در تمامي مسائل بر دو الگوريتم NPGA و VEGA برتري داشته است. و دست آخر آنکه از بين الگوريتم هاي NPGA و VEGA هيچيک برتري مطلق بر ديگري نداشته اند.

(۳-۷). زمينه هاي تحقيقاتي آتي

تحقيقات آتي در ادامه رساله حاضر را مي توان در سه زمينه اصلي: (۱) مدل سازي مسئله طراحي سلول هاي توليدي با تركيب اهداف متفاوت و حل آن با استفاده از روش هاي ديگر بهينه سازي چند هدفي، (۲) استفاده از منطق الگوريتم XGA در حل ساير مسائل بهينه سازي چند هدفي و (۳) توسعه روش هاي حل مسائل بهينه سازي چند هدفي به ويژه الگوريتم هاي ژنتيک چند هدفي ادامه داد.

(۱-۳-۷). توسعه مدل هاي جديد طراحي سلول و حل آنها

در رابطه با مدل سازي مسئله طراحي سلول هاي توليدي همانطور كه نتايج تحقيقات منصوري و همكاران (۲۰۰۰) [۵۳] (پيوست ۴) نشان مي دهد زمينه وسيعي براي توسعه مدلهاي جديد چند هدفي با تركيبات مختلف ورودي ها، معيارها، روش هاي حل و خروجي ها وجود دارد.

مدل هايي با مجموعه هاي جديدي از ورودي هاي در بر دارنده ويژگي هاي خاص محيط هاي مختلف صنعتي مي تواند توسعه يابد. از جمله اين ورودي ها مي توان به مسيرهاي توليد انعطاف پذير و تقاضاي احتمالي قطعات اشاره نمود.

زمينه وسيعي از تحقيقات جديد مي تواند به توسعه مدلهايي با تركيب هاي مختلف معيارهايي كه تاكنون در نظر گرفته نشده اند تخصيص يابد. براي مثال مدل هايي با اهداف كمينه سازي مجموع هزينه ها و بيشينه سازي انعطاف پذيري همراه با ساير اهداف ممكن تاكنون گزارش نشده است. از سوي ديگر نحوه برخورد با اهداف چندگانه نيز مي تواند زمينه ديگري باشد. بيشتر مدلهاي ارائه شده در زمينه طراحي سلول هاي توليدي در حالت چند هدفي از روش وزن دهي براي ادغام اهداف چندگانه استفاده نموده اند. حاصل اين رويه نوعا" يک تابع هدف پيچيده و نامفهوم مي باشد كه حل آن در بهترين حالت تنها يكي از نقاط مجموعه جواب هاي نامغلوب مسئله را پيدا مي كند. استفاده از رويكردهاي ديگري كه بتواند تعداد بيشتري از جواب هاي نامغلوب را در اختيار طراح سيستم قرار دهد نيز مي تواندیكي ديگر از زمينه هاي تحيقاتي باشد.

بررسي امكان استفاده از روش هاي حلي نظير جستجوي ممنوع[۱] ، دسته بندي فازي، و تركيب الگوريتم هاي ژنتيک و شبكه هاي عصبي در حل مدلهاي چند هدفي طراحي سلول نيز از موضوعاتي است كه كار تحقيقاتي بيشتري را طلب مي نمايد.

(۲-۳-۷). حل ساير مسائل بهينه سازي چند هدفي

بسياري از مسائل تصميم گيري و طراحي مهندسي ماهيت چند هدفي دارند. از جمله اين مسائل مي توان به مسئله تعيين توالي عمليات، زمان بندي پروژه ها با محدوديت منابع، طراحي استقرار امكانات و نظاير آنها اشاره نمود كه اهداف و معيارهاي مختلفي در آنها مطرح هستند. استفاده از منطق الگوريتم XGA در حل اين نوع مسائل مي تواند يكي ديگر از زمينه هاي بالقوه تحقيقاتي باشد. در اين رابطه تنظيم پارامترها و كار بر روي اپراتورها ضروري مي نمايد.

(۳-۳-۷). توسعه الگوريتم هاي ژنتيک چند هدفي

توسعه روش هاي حل مسائل بهينه سازي چند هدفي يک زمينه تحقيقاتي جذاب در سالهاي اخير بوده است. مزاياي ذاتي الگوريتم هاي ژنتيک در حل اين نوع مسائل يكي از دلايل اين امر بوده است. كار بر روي توسعه الگوريتم هاي ژنتيک و مقايسه آن با روشهاي موجود در حل مسائل عمومي بهينه سازي چند هدفي نيز به عنوان يک زمينه تحقيقاتي فعال در ادامه رساله حاضر پيشنهاد مي گردد.

---

[۱] Tabu Search

فهرست مراجع

1. AKTURK, M. S. and BALKOSE, H. O., “Part-machine grouping using a multi-objective cluster analysis”, *Int. J. Prod. Res.*, **34**, 2299-2315 (1996)

2. AMIRAHMADI, F. and CHOOBINEH, F., “Identifying the composition of a cellular manufacturing system”, *Int. J. Prod. Res.*, **34**, 2471-2488 (1996)

3. ASKIN, R. G., CRESSWELL, S. H., GOLDBERG, J. B. and VAKHARIA, A., “A Hamiltonian path approach to reordering the part-machine matrix for cellular manufacturing”, *Int. J. Prod. Res.*, **29**, 1081-1100 (1991)

4. BAYKASOGLU, A. and GINDY, N. N. Z., “MOCACEF 1.0: multiple objective capability based approach to form part-machine groups for cellular manufacturing applications, *Int. J. Prod. Res.*, **38**(5), 1133-1161 (2000)

5. BLACK, J. T., “ *The design of the factory with a future*”, McGraw-Hill (1991)

6. BLICKLE, T. and THIELE, L., “A comparison of selection schemes used in genetic algorithms”, TIK report No. 11, Swiss Federal Institute of Technology, Zurich, Switzerland (1995)

7. BOCTOR, F. F., “A linear formulation of the machine-part cell formation problem”, *Int. J. Prod. Res.*, **29**, 343-356 (1991)

8. BOCTOR, F. F., “The minimum cost, machine-part cell formation problem”, *Int. J. Prod. Res.*, **34**, 1045-1063 (1996)

9. BURBIDGE, J. L., “*The introduction of group technology*”, Heineman, London (1975)

10. CAUX, C., BRUNIAUX, R. and PIERREVAL H., “Cell formation with alternative process plans and machine capacity constraints: A new combined approach, *Int. J. Production Economics*, **64**, 279-284 (2000)

11. CHEN, C. Y. and IRANI, S. A., “Cluster first-sequence last heuristics for generating block diagonal forms for a machine-part matrix”, *Int. J. Prod. Res.*, **31**, 2623-2647 (1993)

12. CHEN, W. H. and SRIVASATA, B., “Simulated annealing procedures for forming machine cells in group technology”, *European Journal of Operational Research*, **75**, 100-111 (1994)

13. CHENG, C. H., GOH, C. H. and LEE, A., “Solving the generalized machine assignment problem in group technology”, *Journal of the Operational Research Society*, **47**, 794-802 (1996)

14. CHU, C. -H., “Cluster analysis in manufacturing cellular formation”, *OMEGA: International Journal of Management Science*, **17**, 289-295 (1989)

15. CHU, C. -H, “Manufacturing cell formation by competitive learning”, *Int. J. Prod. Res.*, **31**, 829-843 (1993)

16. CICHOCKI, A. and UNBEHAUEN, R., “*Neural Networks for Optimization and Signal Processing”*, Chichester: John Wiley (1993)

17. COHON, J. L., *Multiobjective Programming and Planning*. New York: Academic Press (1978)

18. DAHEL, N. -E. and SMITH, S. B., “Designing flexibility into cellular manufacturing systems”, *Int. J. Prod. Res.*, **31**, 933-945 (1993)

19. DAITA, S. T. S., IRANI, S. A. and KOTAMRAJU, S.,”Algorithms for production flow analysis”, *Int. J. Prod. Res.*, 37(11), 2609-2638 (1999)

20. DAVIS, L. “Adapting operator probabilities in genetic algorithms”, *Proc. 3rd Intl. Conf. On GA*, June 4-7 1989, p. 61 (1989)

21. DEB, K., “Evolutionary algorithms for multi-criterion optimization”, in engineering design. In Proceedings of Evolutionary Algorithms in Engineering and Computer Science (EUROGEN’99) (1999)

22. DE JONG, K. A., “*An analysis of the behavior of a class of genetic adaptive systems*”, Unpublished PhD Thesis, University of Michigan (1975)

23. FONSECA, C. M. and FLEMING, P. J. “Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization”, In S. For-rest(Ed.), *Proceedings of the Fifth International Conference on Genetic Algorithm*s, San Mateo, California, pp. 416–423. Morgan Kaufmann (1993)

24. GOLDBERG, D. E., “*Genetic Algorithms in Search, Optimization and Machine Learning”*, Reading, MA: Addison-Wesley (1989).

25. GUPTA, Y. P., GUPTA, M. C., KUMAR, A. and SUNDRAM C., “Minimizing total intercell and intracell moves in cellular manufacturing: a genetic algorithm approach”, *International Journal of Computer Integrated Manufacturing*, **8**, 92-101 (1995)

26. GUPTA, Y., GUPTA, M., KUMAR, A. and SUNDARAM C., “A genetic algorithm-based approach to cell composition and layout design problems”, *Int. J. Prod. Res.*, **34**, 447-482 (1996)

27. HAGAN, M. T., DEMUTH, H. B. and BEALE, M., "*Neural Network Design*", Boston, MA: PWS (1996)

28. HAJELA, P. and LIN, C. -Y., "Genetic search strategies in multicriterion optimal design", *Structural Optimization* 4, 99–107 (1992)

29. HO, Y. -C. and MOODIE, C. L., "Solving cell formation problems in a manufacturing environment with flexible processing and routing capabilities", *Int. J. Prod. Res.*, **34**, 2901-2923 (1996)

30. HOLLAND, J. H., " *Adaptation in natural and artificial systems*", Ann Arbor: The University of Michigan Press (1975).

31. HOPFIELD, J. J. and TANK, D. W., "Neural computation of decision in optimization problems", *Biological Cybernetics*, **52**, 141-152 (1985)

32. HORN, J., multicriteria decision making. In T. Back, D. B. Fogel, and Z. Michalewicz (Eds.), *Handbook of Evolutionary Computatio*n. Bristol (UK): Institute of Physics Publishing (1997).

33. HORN, J. and NAFPLIOTIS, N., "Multiobjective optimization using the niched pareto genetic algorithm", IlliGAL Report 93005, Illinois Genetic Algorithms Laboratory, University of Illinois, Urbana Champaign (1993)

34. HORST, R. and PARDALOS, P. M., *Handbook of Global Optimizatio*n. Dor-drecht: Kluwer (1995)

35. HWANG, C. L. and MASUD, A. S. M., *Multiple Objective Decision Making: Methods and Applications* ,Germany: Springer-verlag (1978)

36. HWANG, C. L. and YOON, K., *Multiple Attribute Decisin Making: Methods and applications*, Germany: Springer-verlag (1981)

37. JOINES, J. A., CULBRETH, C. T. and KING, R. E., "Manufacturing cell design: an integer programming model employing genetic algorithms", *IIE Transactions*, **28**, 69-85 (1996)

38. KAO, Y. and MOON, Y. B., "A unified group technology implementation using the backpropagation learning rule of neural networks", *Computers and Industrial Engineering*, **20**, 425-437 (1991)

39. KAPARTHI, S. and SURESH, N. C., "A neural network system for shape-based classification and coding of rotational parts", *Int. J. Prod. Res.*, **29**, 1771-1784 (1991)

40. KAPARTHI, S. and SURESH, N. C., "Machine-component cell formation in group technology: a neural network approach", *Int. J. Prod. Res.*, **30**, 1353-1367 (1992)

41. KAPARTHI, S., SURESH, N. C. and CERVENY, R. P., “An improved neural network leader algorithm for part-machine grouping in group technology”, *European Journal of Operational Research*, **69**, 342-356 (1993)

42. KAZEROONI, M., LOUNG, H. S. and ABHARY, K., “A genetic algorithm based cell design considering alternative routing”, *Computer Integrated Manufacturing Systems*, **10**(2), 93-107 (1997)

43. KING, J. R., and NAKORNCHAI, V., “Machine-component group formation in group technology: review and extension”, *Int. J. Prod. Res.*, **20**, 117-133 (1982)

44. KITAOKA, M, NAKAMURA, R., SERIZAWA, S. and USUKI, J., “Multivariate analysis model for machine-part cell formation problem in group technology”, *Int. J. Production Economics*, **60-61**, 433-438 (1999)

45. KLINCEWICS, J. G. and RAJAN, A., “Using GRASP to solve the component grouping problem”, *Naval Research Logistics*, **41**, 893-912 (1994)

46. KOHONEN, T., “An introduction to neural computing”, *Neural Networks*, **1**, 3-16 (1988)

47. KUSIAK, A., BOE, W. J. and CHENG, C. H., “Designing cellular manufacturing systems: branch and bound and A* approaches”, *IIE Transactions*, **25**, 46-56 (1993)

48. LEE, S. -D. and CHEN, Y. -L., “A weighted approach for cellular manufacturing design: minimizing intercell movement and balancing workload among duplicated machines”, *Int. J. Prod. Res.*, **35**, 1125-1146 (1997)

49. LEE, H. and GARCIA-DIAZ, A., “A network flow approach to solve clustering problems in group technology”, *Int. J. Prod. Res.*, **31**, 603-612 (1993)

50. LIANG, M. and TABOUN, S. M., “Converting functional manufacturing systems into focused machine cells--a bicriterion approach”, *Int. J. Prod. Res.*, **33**, 2147-2161 (1995)

51. LIPPMAN, R. P., “An introduction to computing with neural nets”, *IEEE Acoustics, Speech and Signal Processing Magazine*, **4**, 4-22 (1987)

52. LOGENDRAN, R., “A binary integer programming approach for simultaneous machine-part grouping in cellular manufacturing systems”, *Computers and Industrial Engineering*, **24**, 329-336 (1993)

53. MANSOURI, S. A., MOATTAR-HUSSEINI, S. M. and NEWMAN, S. T., “A review of the modern approaches to multi-criteria cell design”, *Int. J. Prod. Res.*, **38**(5), 1201-1218 (2000)

54. MICHALEWICS, Z. and JANIKOW, “Handling constraints in genetic algorithms”, in *Genetic Algorithms: Proc. 4 th Int. Conf.*, R. K. Belew and L. B. Booker, Eds., San Mateo, CA: Morgan Kaufmann, pp. 151-157 (1991)

55. MIN, H. and SHIN, D., “Simultaneous formation of machine and human cells in group technology: a multiple objective approach”, *Int. J. Prod. Res.*, **31,** 2307-2318 (1993)

56. MOON, C. and KIMN, J., “Genetic algorithm for maximizing the parts flow within cells in manufacturing cell design”, *Computers & Industrial Engineering*, **36**, 379-389 (1999)

57. MOSIER, C. and TAUBE, L., “The facets of group technology and their impacts on implementation--a state-of-the-art survey”, *OMEGA: International Journal of Management Science*, **13**, 381-391 (1985)

58. MUKHOPADHYAY, S. K., SARKAR, P. and PANDA, R. P., “Machine-component grouping in cellular manufacturing by multidimensiontal scaling”, *Int. J. Prod. Res.*, **32**, 457-477 (1994)

59. NAIR, G. J. and NARENDRAN, T. T., “ACCORD: a bicriterion algorithm for cell formation using ordinal and ratio-level data”, *Int. J. Prod. Res.*, **37**(3), 539-556 (1999)

60. OFFODILE, O. F., MEHREZ, A. and GRZNAR, J., “Cellular manufacturing: a taxonomic review framework”, *Journal of Manufacturing Systems*, **13**, 196-220 (1994)

61. PARETO, V. “*Cours D’Economie Politique*”, Volume 1. Lausanne: F.Rouge (1896)

62. PLAQUIN, M. –F and PIERREVAL, H., “Cell formation using evolutionary algorithms with certain constraints”, *Int. J. Production Economics*, **64**, 267-278 (2000)

63. POWEL, D. and SKOLNICK, M. M., “Using genetic algorithms in engineering design optimization with nonlinear constraints”, ”, in *Genetic Algorithms: Proc. 5th Int. Conf.*, S. Forrest, Ed., San Mateo, CA: Morgan Kaufmann, pp. 424-431 (1993)

64. RAJAMANI, D., SINGH, N. and ANEJA, Y. P., “Design of cellular manufacturing systems”, *Int. J. Prod. Res.*, **34**, 1917-1928 (1996)

65. RAMANUJAM, J. and SADAYAPPAN, P., “Optimization by neural networks”, 1988, *Proceedings of the 2nd. IEEE International Conference on Neural Networks*, Vol. **2**, 325-332 (1988)

66. RICHARDSON, J. T., PALMER, M. R., LIEPINS, G. and HILLIARD M., "Some guidelines for genetic algorithms with penalty functions", in *Proc. 3rd. Int. Conf. Genetic Algorithms*, J. D. Schaffer, Ed., San Mateo, CA: Morgan Kaufmann, pp. 191-197 (1989)

67. SAIFUL ISLAM, K. M. and SARKER, B. R., "A similarity coefficient measure and machine-parts grouping in cellular manufacturing systems", *Int. J. Prod. Res.*, 38(3), 699-720 (2000)

68. SANKARAN, S., "Multiple objective decision making approach to cell formation: a goal programming model", *Mathematical and Computer Modeling*, **13**, 71-81 (1990)

69. SANKARAN, S. and KASILINGAM, R. G., "On cell size and machine requirements planning in group technology", *European Journal of Operational Research*, **69**, 373-383 (1993)

70. SCHAFFER, J. D., "Multiple Objective Optimization with Vector Evaluated Genetic Algorithms", *Proc. 1st Int. Conf. Genetic Algorithms and their Applications*, Carnegie-Mellon University, Pittsburgh: USA, 93-100 (1985)

71. SHAFER, S. M. and ROGERS, D. F., "A goal programming approach to the cell formation problem", *Journal of Operations Management*, **10**, 28-43 (1991)

72. SHAFER, S. M., KERN, G. M. and WEI, J. C., "A mathematical programming approach for dealing with exceptional elements in cellular manufacturing", *Int. J. Prod. Res.*, **30**, 1029-1036 (1992)

73. SHANKER, R and VRAT, P., "Some design issues in cellular manufacturing using the fuzzy programming approach", *Int. J. Prod. Res.*, **37**(11), 2545-2563 (1999)

74. SOFIANOPOULOU, S., "Application of simulated annealing to a linear model for the formulation of machine cells in group technology", *Int. J. Prod. Res.*, **35**, 501-511 (1997)

75. SOFIANOPOULOU, S., Manufacturing cells design with alternative process plans and/or replicate machines", *Int. J. Prod. Res.*, **37**(3), 707-720 (1999)

76. SONG, S. and HITOMI, K., "GT cell formation for minimising the intercell parts flow", *Int. J. Prod. Res.*, **30**, 2737-2753 (1992)

77. SRINIVAS, N. and DEB, K., "Multiobjective optimization using non-dominated sorting in genetic algorithms", *Evolutionary Computation* 2(3), 221–248 (1994)

78. STEUER, R. E., *Multiple Criteria Optimization: Theory, Computation, and Application.* New York: Wiley (1986).

79. SU, C. -T. and HSU, C. -M., “Multi-objective machine-part cell formation through parallel simulated annealing”, *Int. J. Prod. Res.,* **36**, 2185-2207 (1998)

80. SURESH, N. C., and KAPARTHI, S., “Performance of fuzzy ART neural network for group technology cell formation”, *Int. J. Prod. Res.*, Vol. 32, No. 7, 1693-1713 (1994)

81. SURESH, N. C., SLOMP, J. and KAPARTHI, S., “The capacitated cell formation problem: a new hierarchical methodology”, *Int. J. Prod. Res.*, **33**, 1761-1784 (1995)

82. TORN, A. and ZILINSKAS, A., “*Global Optimization”,* Berlin: Springer (1989)

83. VENUGOPAL, V. and NARENDRAN, T. T., “A genetic algorithm approach to the machine-component grouping problem with multiple objectives”, *Computers and Industrial Engineering*, **22**, 469-480 (1992a)

84. VENUGOPAL, V. and NARENDRAN, T. T., “Cell formation in manufacturing systems through simulated annealing: An experimental evaluation”, *European Journal of Operational Research*, **63**, 409-422 (1992b)

85. WEI, J. C. and GAITHER, N., “A capacity constrained multiobjective cell formation method”, *Journal of Manufacturing Systems*, **9**, 222-232 (1990)

86. WEMMERLOV, U., and HYER, N. L., “Procedures for the part family/machine group identification problem in cellular manufacturing”, *Journal of Operations Management*, **6**, 125-147 (1986)

87. WEMMERLOV, U. and Hyer, N. L., “Research issues in cellular manufacturing”, *Int. J. Prod. Res.*, Vol. 25, No. 3, 413-431 (1987)

88. WON, Y., “New p-median approach to cell formation with alternative process plans”, Int. J. Prod. Res., **38**(1), 229-240 (2000a)

89. WON, Y., “Two-phase approach to GT cell formation using efficient p-median formulations”, *Int. J. Prod. Res.*, 38(7), 1601-1613 (2000b)

90. ZELENY, M., “*Multiple Criteria Decision Making*”, New York: McGraw-Hill (1982)

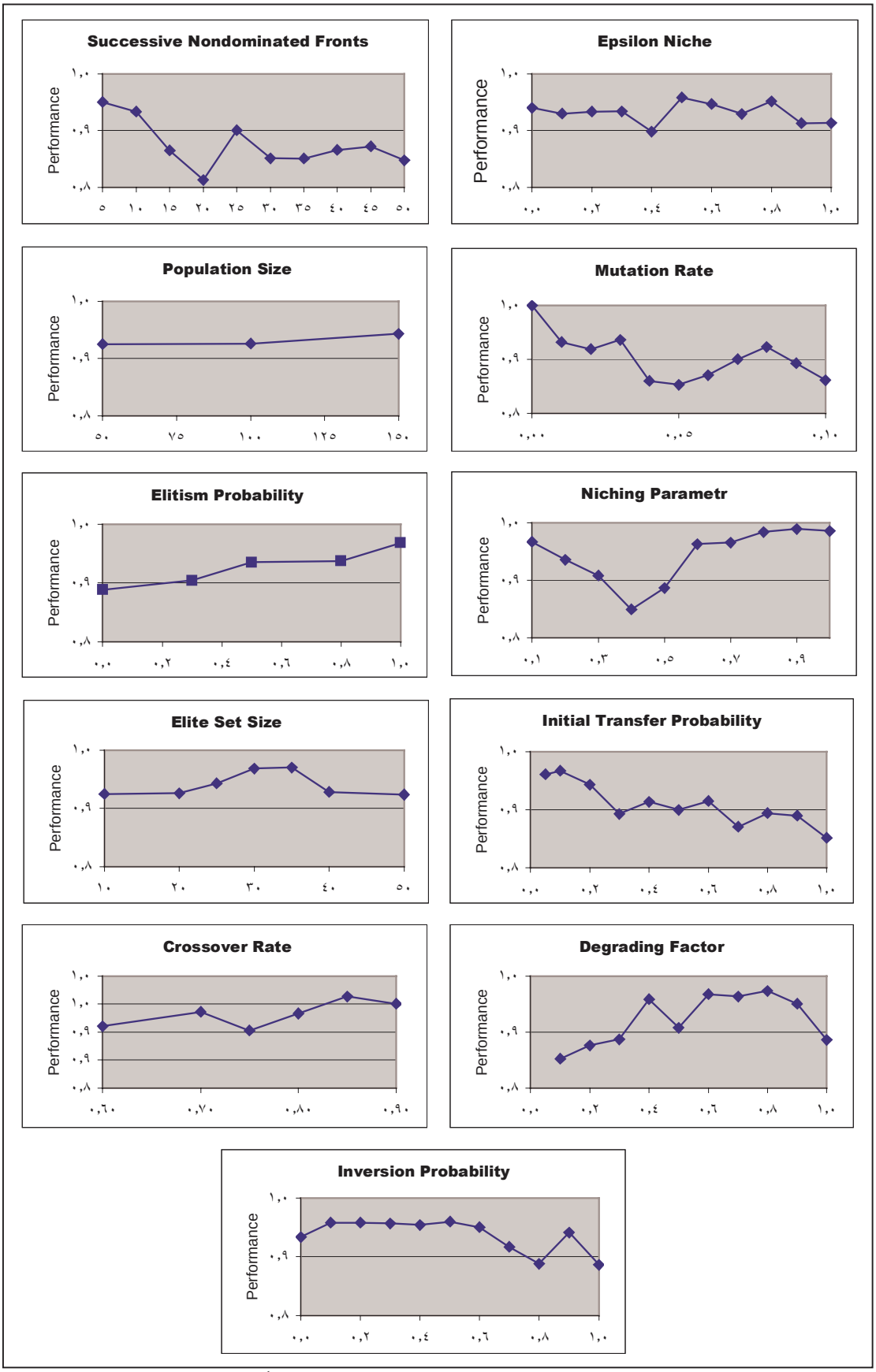
91. ZHAO, C. and WU, Z., “A genetic algorithm for manufacturing cell formation with multiple routes and multiple objectives, *Int. J. Prod. Res.*, **38**(2), 385-395 (2000)

92. ZITZLER, E., “*Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications*”, Ph.D Thesis, Swiss Federal Institute of Technology, Zurich, Switzerland (1999)

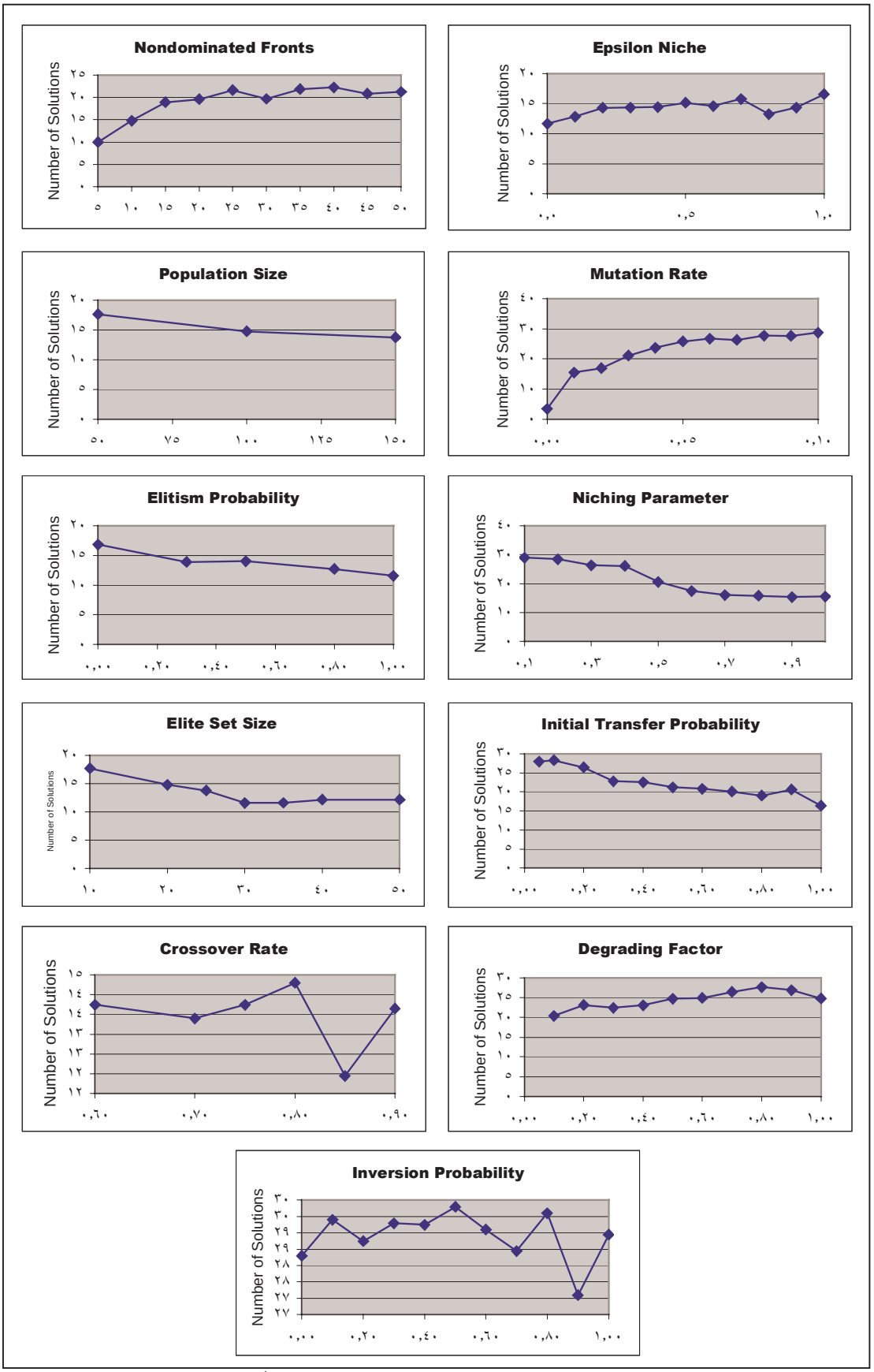
93. ZITZLER, E. and THIELE, L., “An evolutionary algorithm for multiobjective optimization: The Strength Pareto Approach”, TIK report No. 3, Swiss Federal Institute of Technology, Zurich, Switzerland (1998)

پیوست ۱

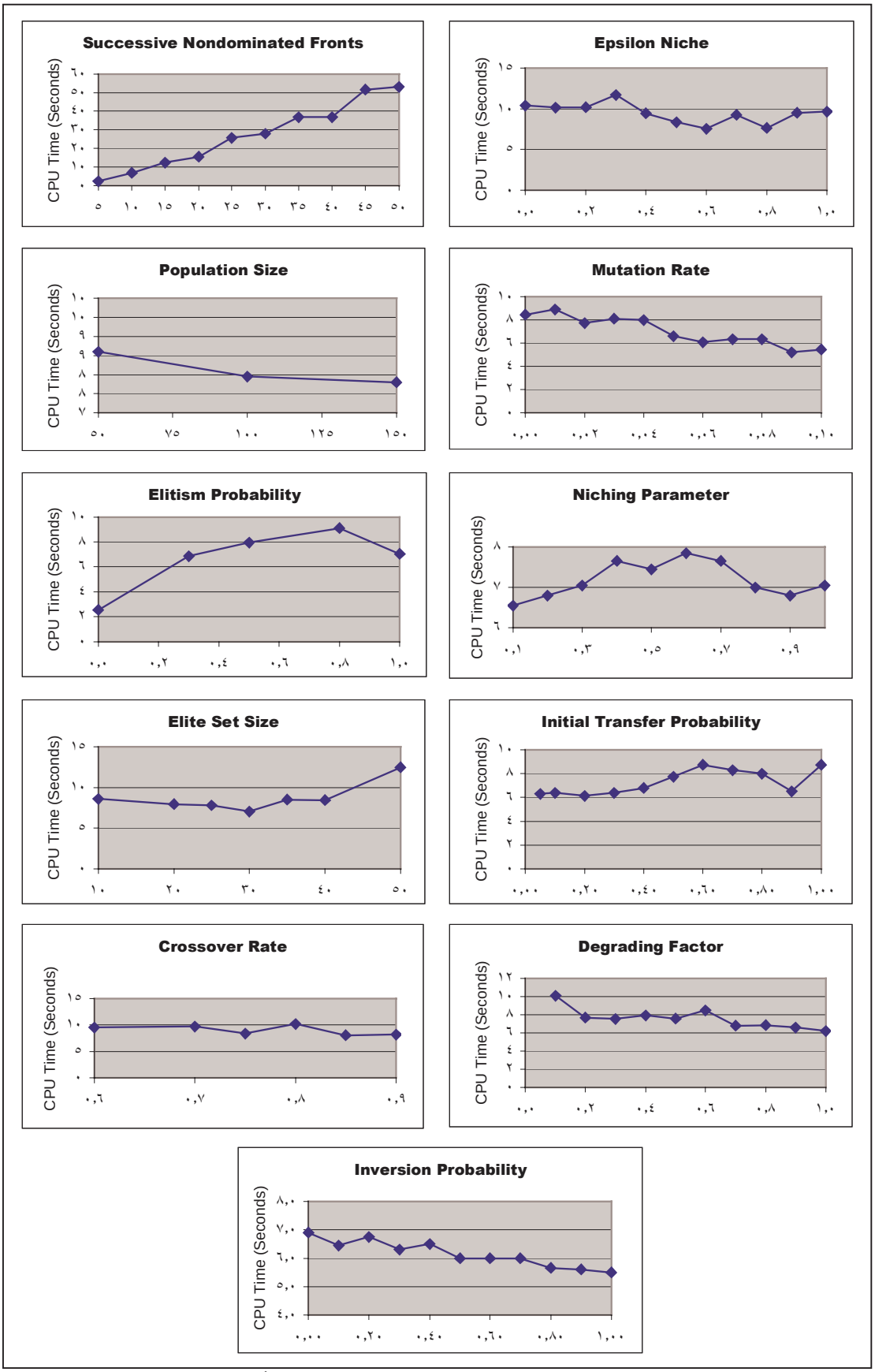
نمودارهاي مربوط به تنظيم پارامترهاي الگوريتم XGA



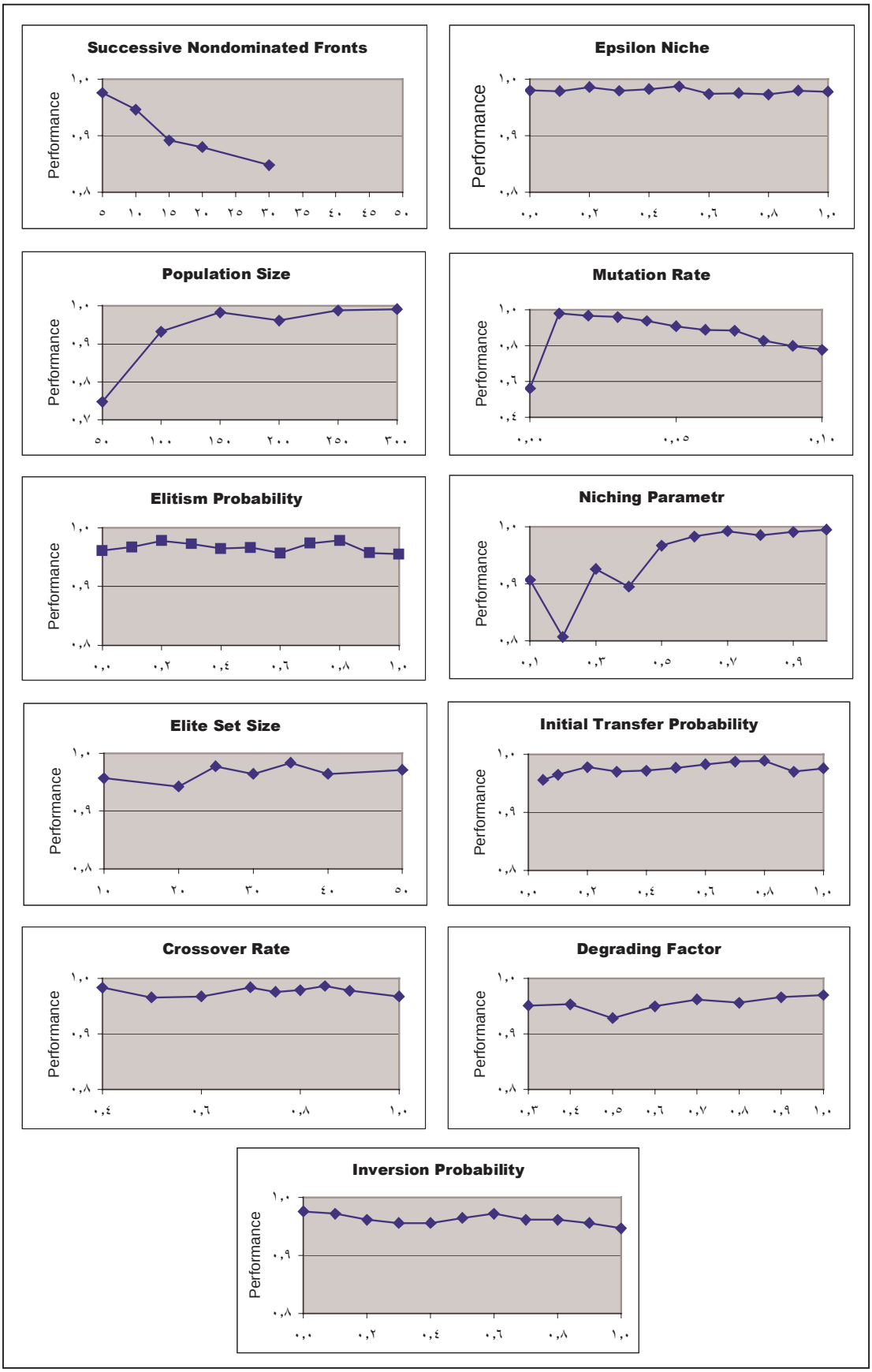
شکل (۱). تغییرات MP_1 نسبت به تغییر پارامترها در مسئله ونوگوپال و نارندران (۱۹۹۲)



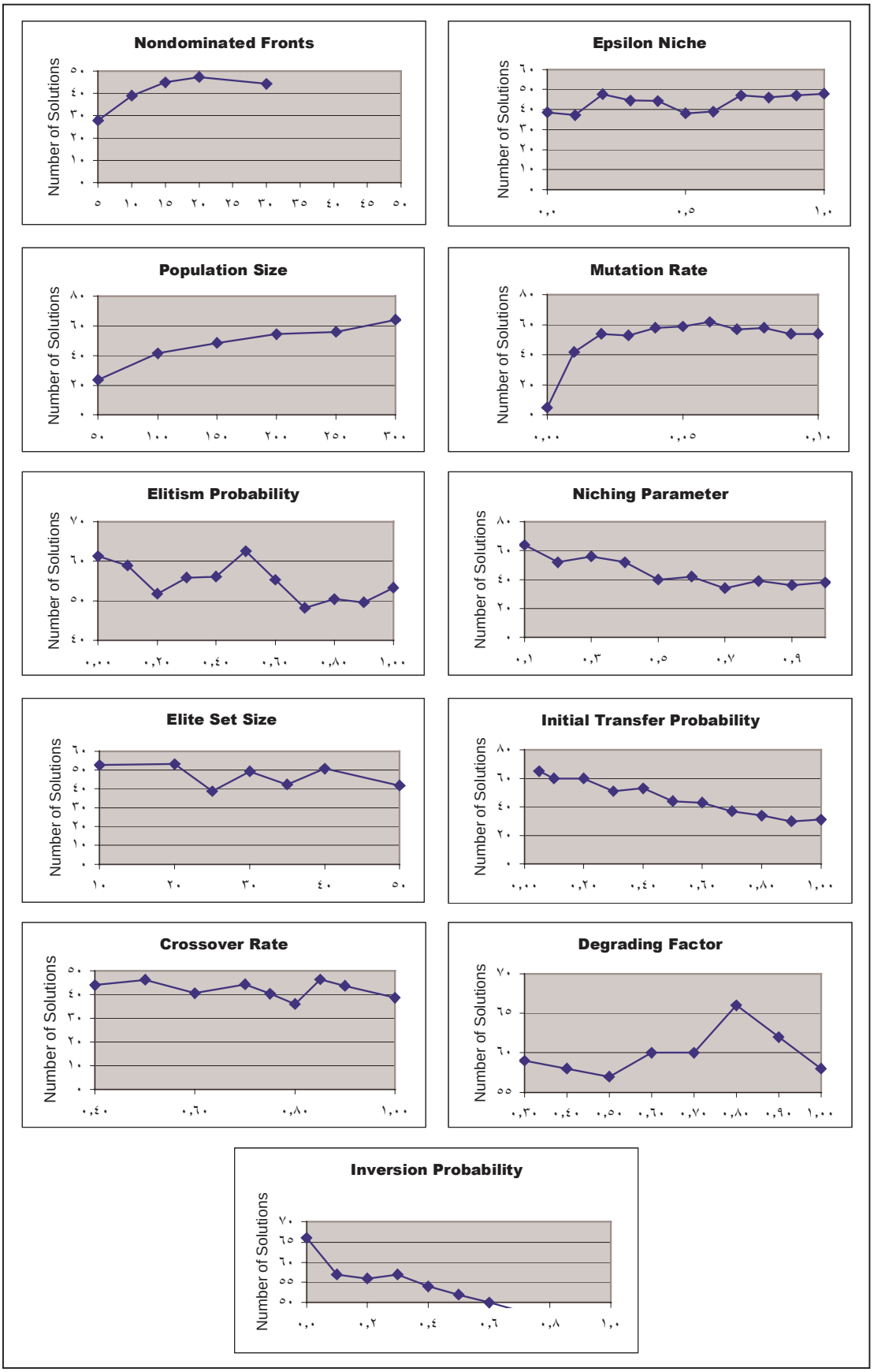
شکل (۲). تغییرات MP_2 نسبت به تغییر پارامترها در مسئله ونوگوپال و نارندران (۱۹۹۲)



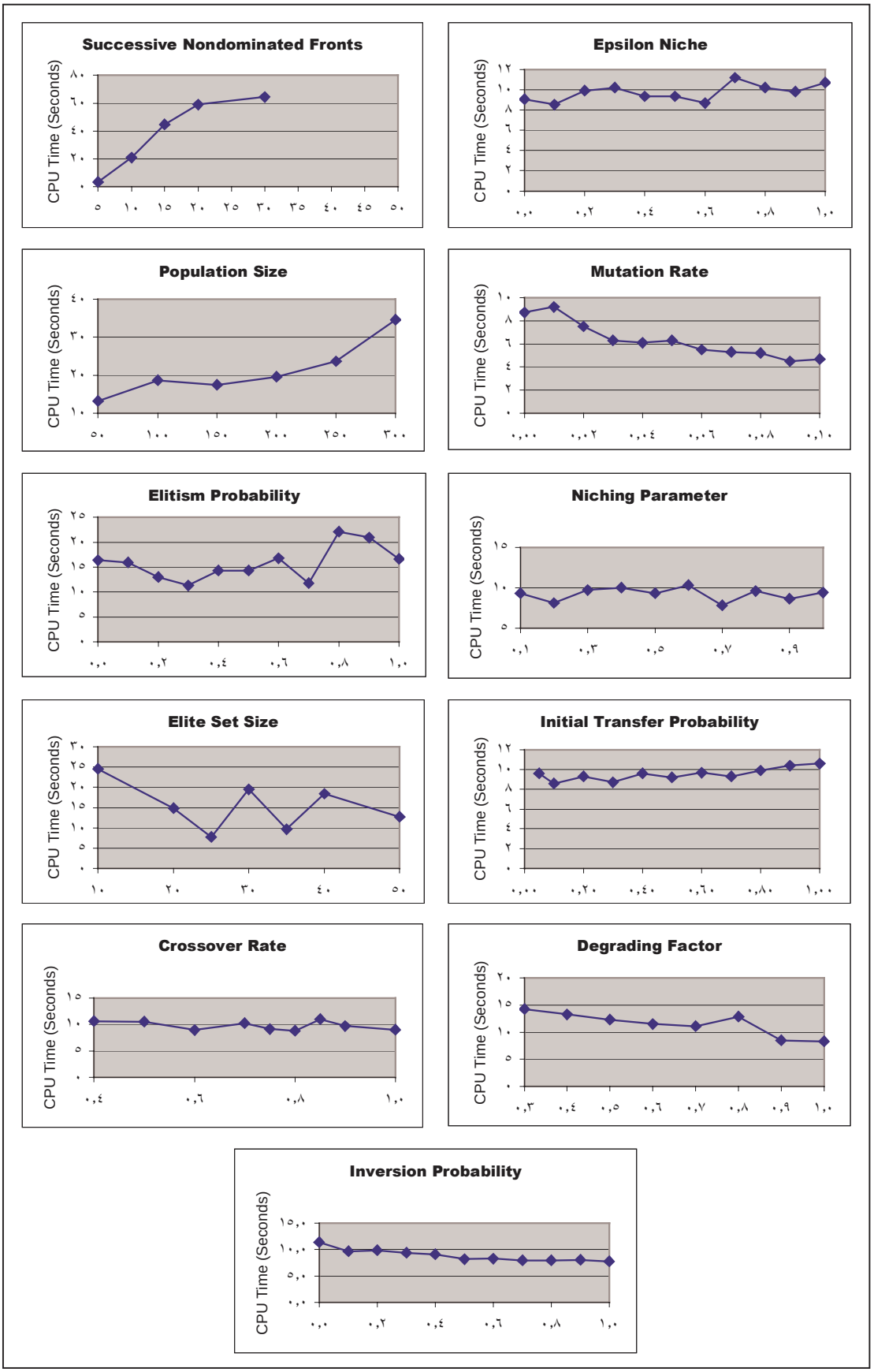
شکل (۳). تغییرات MP_3 نسبت به تغییر پارامترها در مسئله ونوگوپال و نارندران (۱۹۹۲)



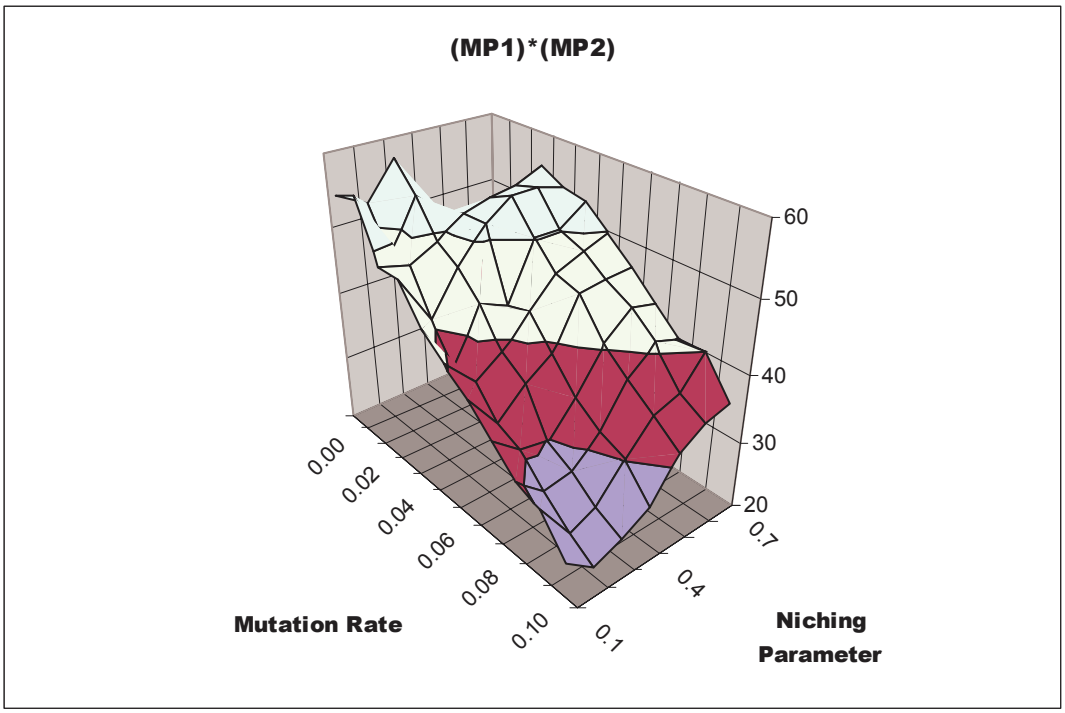
شکل (۴). تغییرات MP_1 نسبت به تغییر پارامترها در مسئله باربیج (۱۹۶۹)



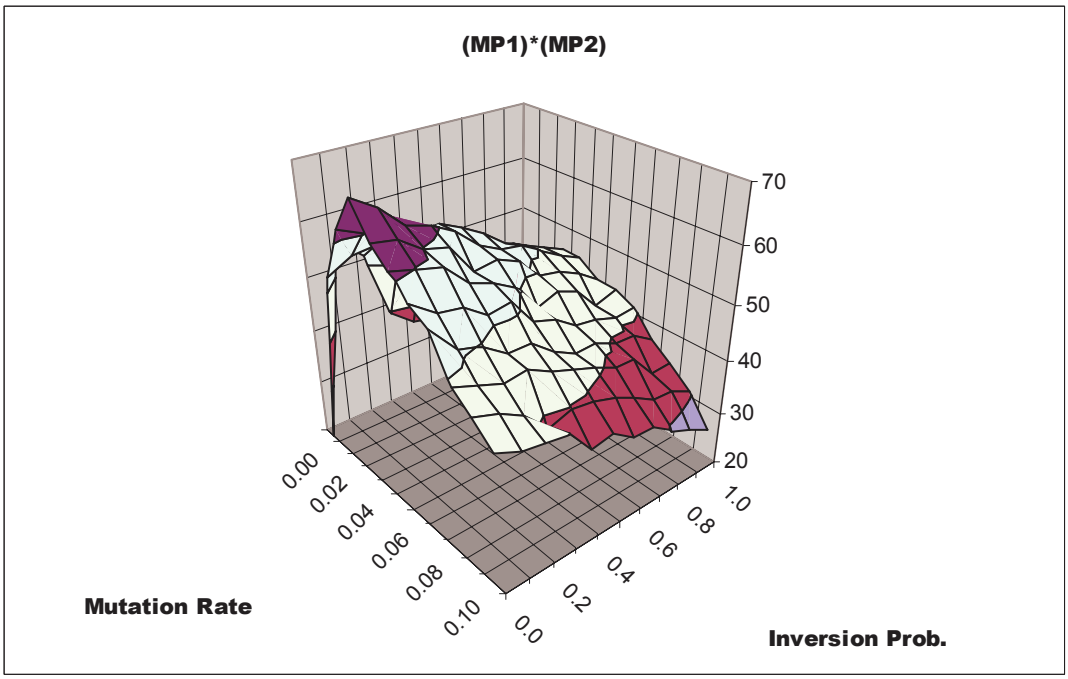
شکل (۵). تغییرات MP_2 نسبت به تغییر پارامترها در مسئله باربیج (۱۹۶۹)



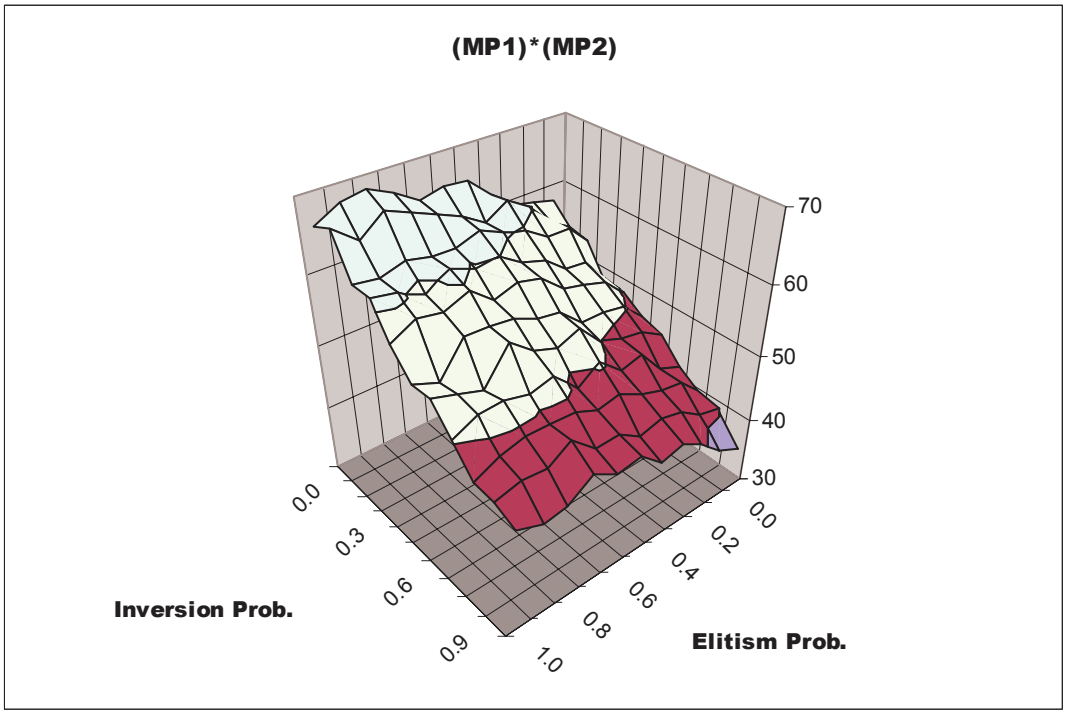
شکل (۶). تغییرات MP_3 نسبت به تغییر پارامترها در مسئله باربیج (۱۹۶۹)



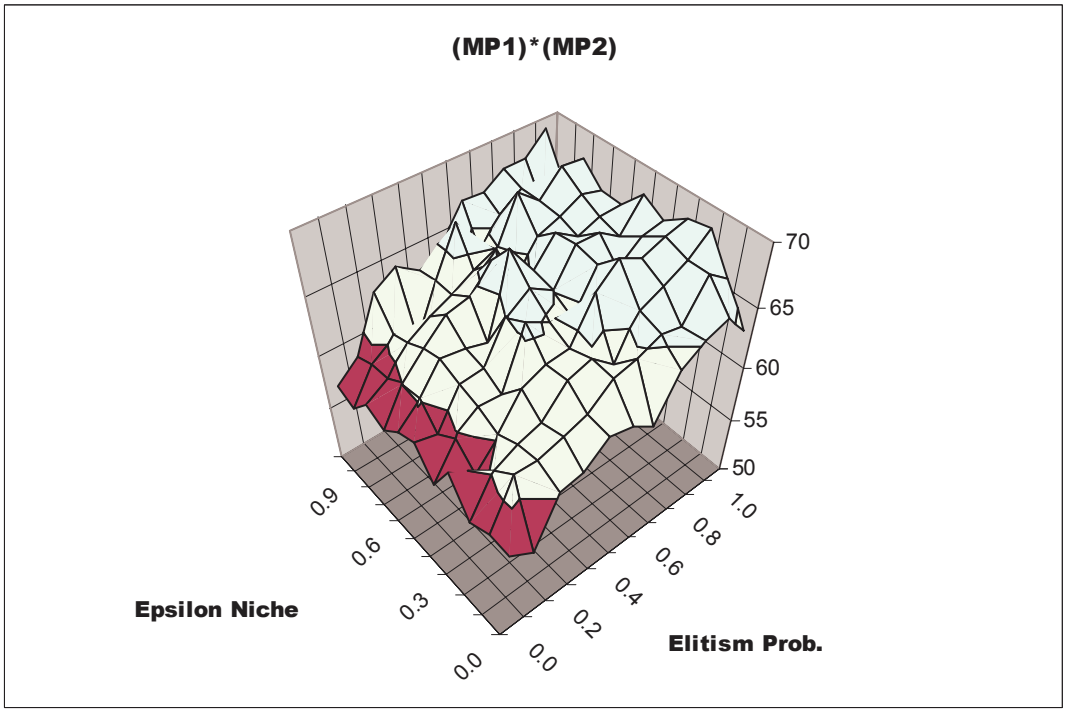
شکل (۷). چگونگي تغييرات ($MP_1 * MP_2$) نسبت به تغييرات
{Niching Parameter}+{Mutation Rate}



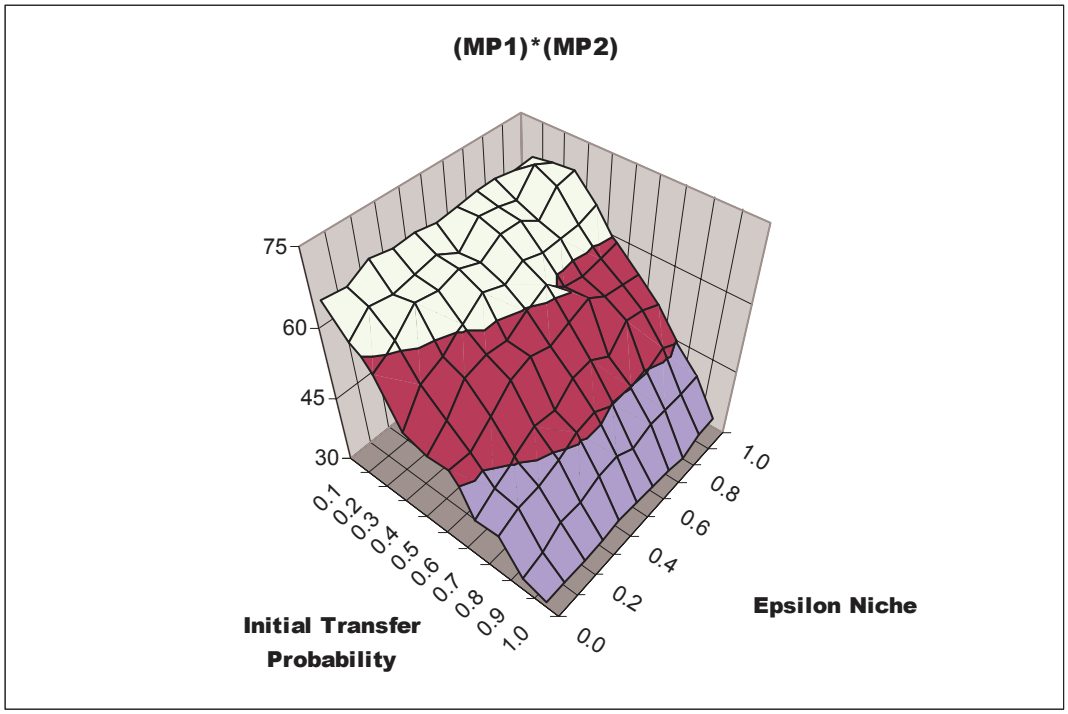
شکل (۸). چگونگي تغييرات ($MP_1 * MP_2$) نسبت به تغييرات
{Mutation Rate}+{Inversion Probability}



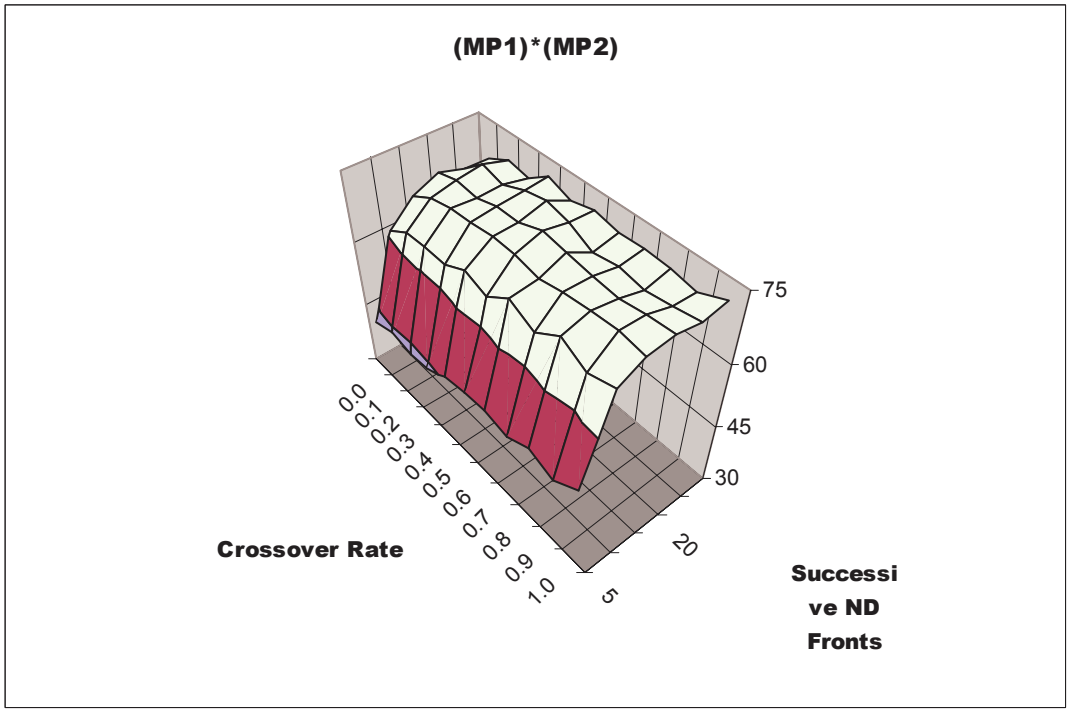
شکل (۹). چگونگي تغییرات ($MP_1 * MP_2$) نسبت به تغییرات
{Inversion Probability}+{Elitism Probability}



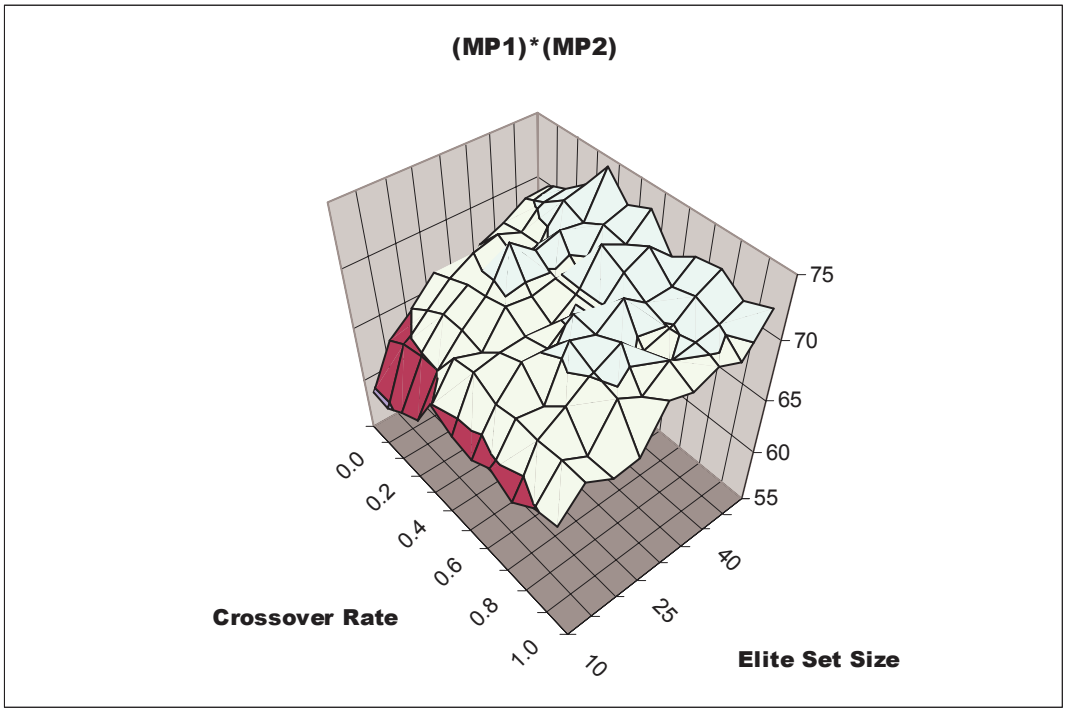
شکل (۱۰). چگونگي تغییرات ($MP_1 * MP_2$) نسبت به تغییرات
{Elitism Probability}+{Epsilon Niche}



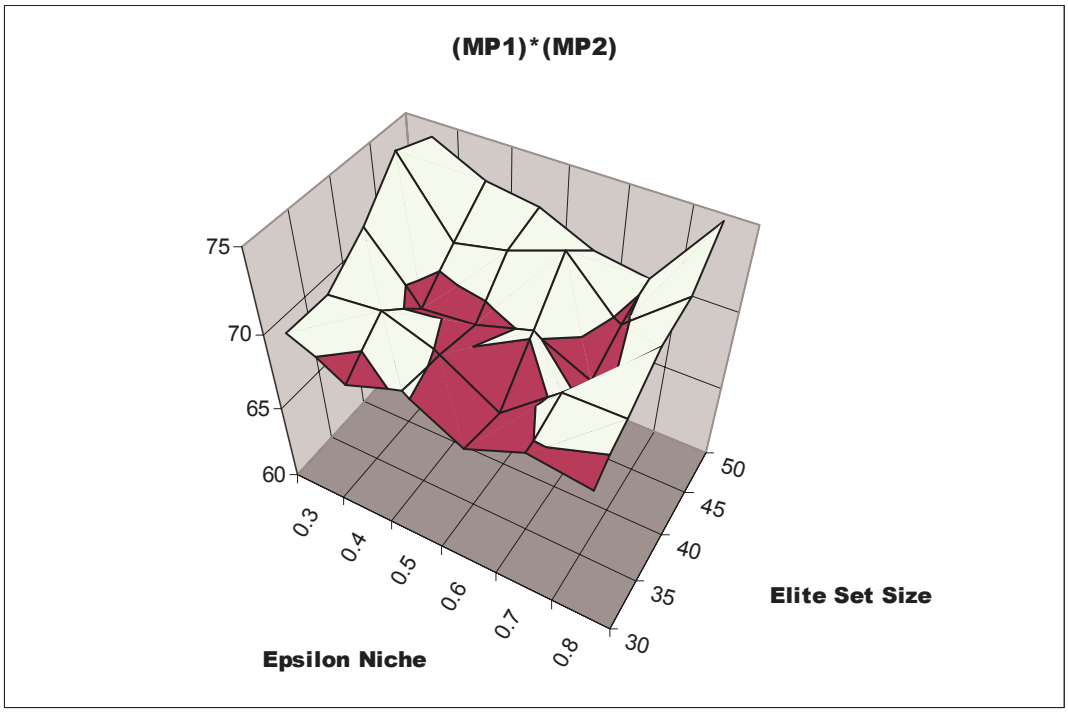
شکل (۱۱). چگونگي تغییرات ($MP_1 * MP_2$) نسبت به تغییرات {Epsilon Niche}+{Initial Transfer Probability}



شکل (۱۲). چگونگي تغییرات ($MP_1 * MP_2$) نسبت به تغییرات {Successive Non-Dominated Fronts}+{Crossover Rate}



شکل (۱۳). چگونگي تغييرات ($MP_1 * MP_2$) نسبت به تغييرات {Crossover Rate}+{Elite Set Size}



شکل (۱۴). چگونگي تغييرات ($MP_1 * MP_2$) نسبت به تغييرات {Elite Set Size}+{Epsilon Niche}

پیوست ۲

متغیرها و برخي توابع نرم افزار XGA

```
/*
==============================================================

        XGA : A Multi-Objective Genetic Algorithm for Cell Design
                            by: S.A.Mansouri
                  Department of Industrial Engineering
                   Amirkabir University of Technology
==============================================================
*/
.
.
.

/*
**********************************
*  Declaration of the variables  *
**********************************
*/
#define random(num)  (rand()%(num))
#define randomize()  srand((unsigned)time(NULL))
#define POPULATION_SIZE  150
#define PCROSS  0.5
#define PMUT  0.03
#define MAX_GEN  250
#define SuccessiveFronts 15
#define SigmaShare  0.6
#define UpperBoundSharedFitness 1.01
#define EliteSet_Size  50
#define InitialTransferProbability 0.1
#define DegradingFactor 0.8
#define EpsilonNiche  0.3
#define P_Elitism  1.0
#define P_Inversion  0.0
#define Max_PartTypes 50
#define Max_MachineTypes 50
#define Max_Cells 20//10
#define CHROM_LENGTH  15
char InputFile[] = {"Bur69_In.txt"}; // The problem to be solved
char UpdatingFile[] = {"Bur69_Ag_FoundFront.out"};
char ReferenceFile[] = {"Bur69_TE_4obj.out"};
#define NondominatedFront_Size 221
char Reference_ND_File[] = {"Bur69_TE_4obj.out"};
time_t StartTime, FinishTime;
double ElapsedTime;
float Sub[Max_PartTypes]; // (Sj)
int Dem[Max_PartTypes]; // (Dj)
int Mch[Max_MachineTypes]; // (Mi)
int PM[Max_PartTypes][Max_MachineTypes]; // PMji's
float t[Max_MachineTypes][Max_PartTypes]; // tij's
int CM[Max_MachineTypes]; // CMi's
int CS[Max_Cells]; // CSk's
int MCS; // Maximum Cell Size
int PartTypes, MachineTypes, Cells;
int ExPartIndex[Max_PartTypes]; // X variables
int ExMachineIndex[Max_MachineTypes]; // i index in the Y variables
int ExCellIndex[Max_Cells]; // k index in indices for the Y variables
int NofXs; // number of the X variables
int NofYs; // Number of the Y variables

class CPopulation
{
public:
        CPopulation(); // Constructor
        double value;
        unsigned char String[CHROM_LENGTH];
        unsigned char OldString[CHROM_LENGTH];
        bool Nondominated;
        bool IgnoredTemporary;
        float DummyFitness;
```

```
        float SharedFitness;
      int NondominatedFront;
        float TransferProbability;
        bool SelectedForEliteSet;
        bool SuitableForMatingSet;
        unsigned char SourceAlgorithm;
        bool Feasibility;
        float fitnessF1;
        float fitnessF2;
        float fitnessF3;
        float fitnessF4;
        unsigned char* ptoX[Max_PartTypes];
        unsigned char* ptoY[Max_MachineTypes][Max_Cells];
        float OU;
        float UC[Max_Cells];
        ~CPopulation(); // Destructor

};

CPopulation::CPopulation() // Constructor
{

}

CPopulation::~CPopulation() // Destructor
{

}

class CExceptionalPart
{
public:
        CExceptionalPart(); // constructor
        int PartIndex;
        int EM_IndexArray[Max_MachineTypes];
        int EM_Size;
        ~CExceptionalPart(); // destructor
};

CExceptionalPart::CExceptionalPart()
{

}

CExceptionalPart::~CExceptionalPart() // constructor
{

}

class CCell
{
public:
        CCell(); //constructor

        int BM[Max_MachineTypes];
        int BM_Size;
        int MC[Max_MachineTypes];
        int MC_Size;
        int HF[Max_PartTypes];
        int HF_Size;
        int GF[Max_PartTypes];
        int GF_Size;
        CExceptionalPart EP[Max_PartTypes];
        CExceptionalPart* ptoEP[Max_PartTypes];
        int EP_Size;
        int EP_IndexArray[Max_PartTypes];
        bool MembershipCheck (int, int*, int);
        int FindRelevantIndexForEP (int);
        ~CCell(); // destructor
};

CCell::CCell() // constructor
{

        for (int j=0; j<Max_PartTypes; j++)
```

```
                ptoEP[j] = &EP[j];
}

bool CCell::MembershipCheck (int mem, int *ptoArray, int ArraySize)
{
        for (int i=0; i<ArraySize; i++)
        {
                 if (mem == *ptoArray)
                      return (true);
                 else
                        ptoArray++;

        }
        return (false);
}

int CCell::FindRelevantIndexForEP (int index)
{
        for (int j=0; j<Max_PartTypes; j++)
        {
                if (index == EP[j].PartIndex)
                       return (j);
        }
        return (-1);
}

CCell::~CCell() // destructor
{

}
struct SIndividual  //for storing individuals in the non-dominated front
{
        unsigned char String[CHROM_LENGTH];
        int value;
};

CPopulation CPool[POPULATION_SIZE]; // objects for the current pool
CPopulation* ptoCPool = CPool; // pointer to the CPool
CPopulation CNewPool[POPULATION_SIZE]; // objects for the new pool
CPopulation CEliteSet[EliteSet_Size];// objects for the Elite Set
CPopulation* ptoCEliteSet = CEliteSet; //pointer to the Elite Set
CPopulation C_NDFrontBaseAlgorithm[POPULATION_SIZE]; //used in computing
//relative performance for large problems
CPopulation* ptoC_NDFrontBaseAlgorithm = C_NDFrontBaseAlgorithm;
#define MaxAntNDFrontSize 10*POPULATION_SIZE //max anticipated size for nondominated
front
CPopulation C_NDFrontPreviousRun[MaxAntNDFrontSize]; //used in updating //nondominated
front of previous run
CPopulation* ptoC_NDFrontPreviousRun = C_NDFrontPreviousRun;
CPopulation C_NDFrontReferenceAlgorithm[MaxAntNDFrontSize]; //used in //computing
relative performance for large problems
CPopulation* ptoC_NDFrontReferenceAlgorithm = C_NDFrontBaseAlgorithm;
CPopulation C_NDFrontCurrentRun[POPULATION_SIZE]; //used in updating //nondominated front
of previous run
CPopulation* ptoC_NDFrontCurrentRun = C_NDFrontCurrentRun;
/* Objects for the prvious generations */
CPopulation C_0th_OldPool[POPULATION_SIZE]; // 0th old generation
CPopulation C_1st_OldPool[POPULATION_SIZE]; // 1st old generation
CPopulation C_2nd_OldPool[POPULATION_SIZE]; // 2nd old generation
CPopulation C_3rd_OldPool[POPULATION_SIZE]; // 3rd old generation
CPopulation C_4th_OldPool[POPULATION_SIZE]; // 4th old generation
/* pointers to the objects of the previous generations */
CPopulation* ptoCOldPool[] = {C_0th_OldPool, C_1st_OldPool,
                              C_2nd_OldPool, C_3rd_OldPool,
                                                  C_4th_OldPool};
CPopulation* selected[POPULATION_SIZE];
CCell MyCell[Max_Cells]; // aray of objects of class CCell
CCell* ptoMyCell = MyCell; // pointer to the first element of the aray
struct SIndividual NondominatedIndividual[NondominatedFront_Size];
SIndividual* ptoNondominatedIndividual= NondominatedIndividual;
// pointer to the array NondominatedIndividual[]
float ExpectedNumber[POPULATION_SIZE];//Used in Rem. Stoch. Samp....
float Distance[POPULATION_SIZE][POPULATION_SIZE]; // Distances
float SharedValue[POPULATION_SIZE][POPULATION_SIZE];// Shared Values
int generations;
```

```
FILE *out_stream;
int parent1;
int parent2;
int child1 ;
int child2;
int index;
float PI;
int Number_ND_Solutions;


************************************************************
*     Some of the Functions of the XGA Genetic Algorithm   *
************************************************************
*/
```

```
/*
<<<<<<<<<<<<<<<<<<<<F١-F١-F١>>>>>>>>>>>>>>>>>>>>>>

        Min F١: Intercellular movements

  With fitness function: NorFactor/(NorFactor+f١)

==================================================
*/
float evaluateF١(int x)

{

RelateStringtoVariables(x);

int k, j, i;
float f١ = ٠;

int NorFactor = ١٠٠٠٠;


        for (k=٠; k<Cells; k++) // For Xj's

{

for (j=١; j<=PartTypes; j++)

{

if ((ptoMyCell + k)->MembershipCheck (j, (ptoMyCell + k)->EP_IndexArray,
(ptoMyCell + k)->EP_Size)==true)
{
int IndexForEP = (ptoMyCell + k)->FindRelevantIndexForEP (j);
// determines the relevant index for the EParray
       for (i=١; i<=MachineTypes; i++)

{
//if (ptoEM[k]->MemberCheck(i,
ptoEM[k]->iArray)==true)
if ((ptoMyCell + k)->MembershipCheck
(i,
(ptoMyCell + k)->ptoEP[IndexForEP]->EM_IndexArray, (ptoMyCell + k)->



ptoEP[IndexForEP]->EM_Size) == true)

f1 += (1-*CPool[x].ptoX[j-1])*(1-*CPool[x].ptoY[i-1][k])*PM[j-1][i-1];
            // [i-١] refers to the place


 } // end of the i loop

     } // end of if
}  // end of the j loop

} // end of the k loop

   return(NorFactor/(NorFactor+f١));

}
```

```
/*
<<<<<<<<<<<<<<<<<<<<F2-F2-F2>>>>>>>>>>>>>>>>>>>>>>>
                      Min F۲: Cost

  With fitness function: NorFactor/(NorFactor+f۲)

==================================================
*/
float evaluateF۲(int x)
{

RelateStringtoVariables(x);

int k, j, i;
float f۲ =۰;

int NorFactor = ۱۰۰۰۰۰;

    /* Calculations based on the exceptional parts (Xj's) */

        for (k=۰; k<Cells; k++) // For Xj's
{

for (j=۱; j<=PartTypes; j++)
{

//if (ptoEP[k]->MemberCheck(j, ptoEP[k]->iArray)==true)
if ((ptoMyCell + k)->MembershipCheck (j,
(ptoMyCell + k)->EP_IndexArray,
(ptoMyCell + k)->EP_Size)==true)
{
   f۲ += *CPool[x].ptoX[j-۱]*Dem[j-۱]*Sub[j-۱];  // j-۱
referrs to the place

   int IndexForEP = (ptoMyCell + k)->FindRelevantIndexForEP
(j);
   // determines the relevant index for the EP array
       for (i=۱; i<=MachineTypes; i++)
   {
if ((ptoMyCell + k)->MembershipCheck (i,(ptoMyCell + k)->
ptoEP[IndexForEP]->EM_IndexArray,
(ptoMyCell + k)->ptoEP[IndexForEP]->EM_Size) == true)

f۲ += *CPool[x].ptoY[i-
۱][k]*Mch[i-۱]; // [i-۱] refers to the place

                                        } // end of the i loop

} // end of if




}  // end of the j loop

} // end of the k loop

   return(NorFactor/(NorFactor+f۲));

       }
```

```
/*
<<<<<<<<<<<<<<<<<<<<F۳-F۳-F۳>>>>>>>>>>>>>>>>>>>>>>>
```

```
            Min F۳: Underutilisation
  With fitness function: NorFactor/(NorFactor+f۳)
==================================================
*/
float evaluateF۳(int x)
{
RelateStringtoVariables(x);
int k, j, i;
float f۳ = ۰;
float Nominator۱;
float Denominator۱;
float Nominator۲;
float Denominator۲;
int NorFactor = ۱;
/* Calculating utilisation of cells (UCk's) */
    for (k=۰; k<Cells; k++)
{
Nominator۱ = ۰;
Denominator۱ = ۰;
              // Calculation for the nominator of UCk
for (i=۱; i<=MachineTypes; i++)
{
   //if (ptoMC[k]->MemberCheck(i, ptoMC[k]->iArray) == true)
if ((ptoMyCell + k)->MembershipCheck (i,
(ptoMyCell + k)->MC,
(ptoMyCell + k)->MC_Size)==true)
{
for (j=۱; j<=PartTypes; j++)
{
//if (ptoHF[k]->MemberCheck(j, ptoHF[k]-
>iArray)==true)
if ((ptoMyCell + k)->MembershipCheck (j,
(ptoMyCell + k)->HF,
(ptoMyCell + k)->HF_Size)==true)

Nominator۱ += Dem[j-۱]*t[i-۱][j-۱];

} // end of the ۱st j loop

for (j=۱; j<=PartTypes; j++)
{
//if (ptoEP[k]->MemberCheck(j, ptoEP[k]-
>iArray)==true)
if ((ptoMyCell + k)->MembershipCheck (j,
(ptoMyCell + k)->EP_IndexArray,
(ptoMyCell + k)->EP_Size)==true)

Nominator۱ -= *CPool[x].ptoX[j-
۱]*Dem[j-۱]*t[i-۱][j-۱];

} // end of the ۲nd j loop

for (j=۱; j<=PartTypes; j++)
```

```
{
//if (ptoGF[k]->MemberCheck(j, ptoGF[k]-
>iArray)==true)
if ((ptoMyCell + k)->MembershipCheck (j,
(ptoMyCell + k)->GF,
(ptoMyCell + k)->GF_Size)==true)

Nominator۱ += (۱-*CPool[x].ptoX[j-
۱])*Dem[j-۱]*t[i-۱][j-۱];

   } // End of the ۳rd j loop
} // end of the i's if loop

   // Calculation for the denominator of UCk
//if (ptoMC[k]->MemberCheck(i, ptoMC[k]->iArray)==true)
if ((ptoMyCell + k)->MembershipCheck (i,
(ptoMyCell + k)->MC,
(ptoMyCell + k)->MC_Size)==true)

Denominator۱ += CM[i-۱];

   //if (ptoBM[k]->MemberCheck(i, ptoBM[k]->iArray)==true)
if ((ptoMyCell + k)->MembershipCheck (i,
(ptoMyCell + k)->BM,
(ptoMyCell + k)->BM_Size)==true)

 Denominator۱ += *CPool[x].ptoY[i-۱][k]*CM[i-
۱];


 } // end of the i loop

CPool[x].UC[k] = Nominator۱/Denominator۱; // Utilisation of cells

} // end of the k loop


/* Calculating total utilisation */
Nominator۲ = ۰;
Denominator۲ = ۰;

for (k=۰; k<Cells; k++)
{
Nominator۲ += CPool[x].UC[k]*CS[k];

for (i=۱; i<=MachineTypes; i++)
{


if ((ptoMyCell + k)->MembershipCheck (i,
(ptoMyCell + k)->BM,
(ptoMyCell + k)->BM_Size)==true)

 Nominator۲ +=
CPool[x].UC[k]**CPool[x].ptoY[i-۱][k];

 } // end of the i loop for Nominator۲

Denominator۲ += CS[k];
```

```
for (i=۱; i<=MachineTypes; i++)
{
   //if (ptoBM[k]->MemberCheck(i, ptoBM[k]->iArray)==true)
if ((ptoMyCell + k)->MembershipCheck (i,
(ptoMyCell + k)->BM,
(ptoMyCell + k)->BM_Size)==true)

Denominator۲ += *CPool[x].ptoY[i-۱][k];

} // end of the i loop for Denominator۲

} // end of the k loop

CPool[x].OU =  (Nominator۲/Denominator۲); // Overall utilisation

f۳ = ۱-CPool[x].OU; // Overall Underutilisation

   return(NorFactor / (NorFactor + f۳));

       }
```

```
/*
<<<<<<<<<<<<<<<<<<<<F٤-F٤-F٤>>>>>>>>>>>>>>>>>>>>>
            Min F٤: Load unbalance
  With fitness function: NorFactor/(NorFactor+f٤)
==================================================
*/
float evaluateF٤(int x)
{
RelateStringtoVariables(x);
int k;
float Nominator = ٠;
float f٤;
int NorFactor = ۱;
for (k=0; k<Cells; k++) //
Nominator += (float) pow((CPool[x].UC[k] - CPool[x].OU), ۲);

f٤ = (Nominator/(Cells - ۱));

return (NorFactor/(NorFactor + f٤));
}
```

```
/*
=========================================================
                DoNondominatedSorting
     performs nondominated sorting untill all the
  individuals have been assigned shared fitness values
=========================================================
*/
void DoNondominatedSorting()
{
       int Condition = 1; // for execution of the (do...while) loop
       int Front = 1; // refers to the nondominated front
```

```
        PrintGenerationNumber(out_stream); /* prints generation number in the file
MOCDx_Gen.out */

        SetInitialValues();

        do
        {
                Condition = 1;
                MarkDominatedSolutions();

                PrintCurrentFront(out_stream, Front);

                AssignDummyFitness();
                CalculateHammingDistances();
                CalculateSharingValues();
                CalculateSharedFitnessValues();
                IgnoreCurrentNondominatedFront();
                ReviveDominatedIndividuals();

                for (int i=0; i<POPULATION_SIZE; i++)
                        // to verify if there is any individual need to be checked
                {
                if ((ptoCPool + i)->IgnoredTemporary == false)
                        Condition = 0;

                }

        Front++;

        }

        while (Condition == 0);
         /* continues untill all of the individuals
                   have been marked as IgnoredTemporary */
}
```

```
/*
============================================================
                     DominatedCheck
 investigates either i or j (or none of them) is dominated
============================================================
*/
char DominatedCheck(int i, CPopulation* ptoG1, int j, CPopulation* ptoG2)
{
CPopulation* ptoi = ptoG1 + i; // pointer to the CPool[i] in generation G1
CPopulation* ptoj = ptoG2 + j; // pointer to the CPool[j] in generation G2

float F1i = ptoi->fitnessF1;
float F2i = ptoi->fitnessF2;
float F3i = ptoi->fitnessF3;
float F4i = ptoi->fitnessF4;

float F1j = ptoj->fitnessF1;
float F2j = ptoj->fitnessF2;
float F3j = ptoj->fitnessF3;
float F4j = ptoj->fitnessF4;


         if ((F1i> F1j && F2i> F2j && F3i> F3j && F4i>=F4j) ||
                        (F1i> F1j && F2i> F2j && F3i==F3j && F4i>=F4j) ||
                        (F1i> F1j && F2i==F2j && F3i> F3j && F4i>=F4j) ||
                        (F1i> F1j && F2i==F2j && F3i==F3j && F4i>=F4j) ||
                        (F1i==F1j && F2i> F2j && F3i> F3j && F4i>=F4j) ||
                        (F1i==F1j && F2i> F2j && F3i==F3j && F4i>=F4j) ||
                        (F1i==F1j && F2i==F2j && F3i> F3j && F4i>=F4j) ||
                        (F1i==F1j && F2i==F2j && F3i==F3j && F4i> F4j))

                       return ('S'); // j is dominated, 'S' stands for "SecondElement"
```

```
        else if ((F1i==F1j && F2i==F2j && F3i==F3j && F4i< F4j) ||
                        (F1i==F1j && F2i==F2j && F3i< F3j && F4i<=F4j) ||
                        (F1i==F1j && F2i< F2j && F3i==F3j && F4i<=F4j) ||
                        (F1i==F1j && F2i< F2j && F3i< F3j && F4i<=F4j) ||
                        (F1i< F1j && F2i==F2j && F3i==F3j && F4i<=F4j) ||
                        (F1i< F1j && F2i==F2j && F3i< F3j && F4i<=F4j) ||
                        (F1i< F1j && F2i< F2j && F3i==F3j && F4i<=F4j) ||
                        (F1i< F1j && F2i< F2j && F3i< F3j && F4i<=F4j))

                      return ('F'); // i is dominated, 'F' stands for the "FirstElement"


        return ('N');
}
```

```
/*
==================================================
               AssignDummyFitness
  assigns dummy fitness values to individuals
           in the nondominated front
==================================================
*/
void AssignDummyFitness()
{
        for (int i=0; i<POPULATION_SIZE; i++)
        {
                if (((ptoCPool + i)->IgnoredTemporary == false) &&
                        ((ptoCPool + i)->Nondominated == true))

                           (ptoCPool + i)->DummyFitness =
                                 float (MinSharedFitness() * 0.99);
                                         //(MinSharedFitness());
  /* multiplier 0.99 is for keeping the dummy fitness value
               less than the minimum shared fitness so far */
        }
}
```

```
/*
==============================================
        CalculateSharedFitnessValues
      calculates shared fitness values
      in the current nondominated front
==============================================
*/
void CalculateSharedFitnessValues()
{
        for (int i=0; i<POPULATION_SIZE; i++)
        {
         if (((ptoCPool + i)->IgnoredTemporary == false) &&
                 ((ptoCPool + i)->Nondominated == true))
                 /* which means i is in the current nondominated front */
                 (ptoCPool + i)->SharedFitness =
                    ((ptoCPool + i)->DummyFitness) / NicheCount(i);
        }
}
```

```
/*
==================================================
               CalculateNicheCount
  calculates niche count of a given individual
        in the current nondominated front
==================================================
*/
float NicheCount(int i)
{
        float Counter = 0;

        for (int j=0; j<POPULATION_SIZE; j++)
        {
                if (((ptoCPool + j)->IgnoredTemporary == false) &&
                        ((ptoCPool + j)->Nondominated == true))
```

```
                /* which means j is in the current nondominated front */
                        Counter += SharedValue[i][j];
        }

        return (Counter);
}
```

```
/*
=====================================================
                UpdateEliteSet
     updates reference set by adding nondominated
            solutions of the current generation
=====================================================
*/
void UpdateEliteSet(int GenerationNumber, FILE* Stream1)
{
        int i, j, k, t;
        int MatingSetIndex = 0; // to be used as the index for individuals in the
MatingSet
        char DominatedIndividual = 'N'; // stands for "None"
        bool AllIndividualsHaveBeenSelected; // to exit the loop after all individuals
have been selected

        CPopulation *ptoMatingSet = new CPopulation [POPULATION_SIZE + EliteSet_Size + 1];
        if (GenerationNumber == 1)
        {
                int TempIndex = 0; // temporary index for the members of EliteSet
                for (int i=0; i<POPULATION_SIZE; i++)
                {
                        if ((ptoCPool+i)->NondominatedFront == 1 &&
                                (ptoCPool+i)->Feasibility == true)
                        {
                                *(ptoCEliteSet + TempIndex) = CPool[i];
                                TempIndex++;
                        }
                }
        }

        for (i=0; i<POPULATION_SIZE; i++)
        {
                if (((ptoCPool+i)->NondominatedFront == 1) &&
                        ((ptoCPool+i)->SuitableForMatingSet != false) &&
                        ((ptoCPool+i)->Feasibility == true))
                {
                        j=0;

                        bool i_IsNondominated = true; // to check if 'i' is nondominated

                        while ((i_IsNondominated) && j<EliteSet_Size)
                        {
                                if ((ptoCEliteSet + j)->Nondominated == true); // to ignore
checking domimated j's
                                {
                                        DominatedIndividual = DominatedCheck(i, ptoCPool, j,
ptoCEliteSet);

                                        switch (DominatedIndividual)
                                        {

                                                case 'F': // First element (i) is dominated
                                                {
                                                        (ptoCPool + i)->SuitableForMatingSet
= false;
                                                        i_IsNondominated = false;
                                                }

                                                case 'S': // Second element (j) is dominated
                                                        (ptoCEliteSet + j)->Nondominated =
false;

                                                break;
                                        }
```

```
                                }

                                j++;

                        } // end of the 'while' loop



                } // end of the 'if' loop
        } // end of the 'i' loop
        /* copying nondominated solutions of the previous Elite Set */
        for (j=0; j<EliteSet_Size; j++)
        {
                if ((ptoCEliteSet + j)->NondominatedFront == 1 &&
                        ((ptoCEliteSet + j)->Nondominated != false))
                {
                        //MatingSet[MatingSetIndex] = CEliteSet[j];
                        *(ptoMatingSet + MatingSetIndex) = CEliteSet[j];
                        MatingSetIndex++;
                }
        }


        /* copying nondominated solutions of the current generation */
        for (i=0; i<POPULATION_SIZE; i++)
        {
                if (((ptoCPool + i)->NondominatedFront == 1) &&
                        ((ptoCPool + i)->SuitableForMatingSet == true) &&
                        ((ptoCPool + i)->Feasibility == true))
                {
                        *(ptoMatingSet + MatingSetIndex) = CPool[i];
                        MatingSetIndex++;
                }
        }
        for (j=0; j<(POPULATION_SIZE + EliteSet_Size); j++)
        {
                //if ((ptoMatingSet + j)->NondominatedFront == 1) // to ensure that empty
records are skipped
                //{
                        //for (k=(j+1); k<(POPULATION_SIZE + EliteSet_Size); k++)
                k=(j+1);
                while (k < (POPULATION_SIZE + EliteSet_Size))
                {
                        //if ((ptoMatingSet + k)->NondominatedFront == 1) // to ensure that
empty records are skipped
                        //{
                        if (((ptoMatingSet + j)->value == (ptoMatingSet + k)->value) &&
                                ((ptoMatingSet + k)->NondominatedFront == 1))
                        {
                                for (t=k; t<(POPULATION_SIZE + EliteSet_Size); t++)
                                        *(ptoMatingSet + t) = *(ptoMatingSet + (t+1));
                                        // to pull all individuals one position up

                                MatingSetIndex--;
                                k--;

                        } // end of the 'if' loop
                                //}// end of the inner 'if' loop
                        k++;
                } // end of the 'k' loop
                //} // end of the outer'if' loop

        } // end of the 'j' loop

        if (MatingSetIndex < EliteSet_Size) // number of individuals doesn't exeed
EliteSet_Size
        {
          for (j=0; j<EliteSet_Size; j++)
                  //CEliteSet[j] = MatingSet[j];
                  CEliteSet[j] = *(ptoMatingSet + j);
        }

        else
        {
```

```
                /* Initialization of the Mating Set*/
                for (k=0; k<(POPULATION_SIZE + EliteSet_Size); k++)
                {
                        (ptoMatingSet + k)->SelectedForEliteSet = false;
                        (ptoMatingSet + k)->TransferProbability = float
(InitialTransferProbability);
                }

                int EliteSetIndex = 0; // Index of individuals in the Reference Set

                do
                {
                   /* to exit the loop after all individuals have been selected */
                   AllIndividualsHaveBeenSelected = true;

                   for (i=0; i<(MatingSetIndex - 1); i++)
                   {
                           if ((ptoMatingSet + i)->SelectedForEliteSet == false)
                           {
                                       AllIndividualsHaveBeenSelected = false;

                                       if (flip ((ptoMatingSet + i)->TransferProbability))
                                       {
                                               CEliteSet[EliteSetIndex] = *(ptoMatingSet +
i);
                                               (ptoMatingSet + i)->SelectedForEliteSet =
true;
                                               DegradeTransferProbabilities(i,
MatingSetIndex, ptoMatingSet);
                                               EliteSetIndex++;
                                       } // end of the inner 'if' loop
                           } // end of the outer 'if' loop

                   } // end of the 'i' loop

                } // end of the 'do...while' loop
                while (AllIndividualsHaveBeenSelected == false);
                //while (EliteSetIndex < EliteSet_Size);


        } // end of 'else' loop

        delete [] ptoMatingSet;

        fprintf (Stream1, "\n\n*** Reference Set in Generation [%d] ***",
GenerationNumber);
        fprintf (Stream1, "\n\nDecoded Individuals");
    fprintf (Stream1, "\n-------------------");

        for (j=0; j<EliteSet_Size; j++)
        {
                if ((ptoCEliteSet + j)->NondominatedFront == 1) // to ensure that the
empty members are not printed
                        fprintf (Stream1, "\n    %20.0f", (ptoCEliteSet + j)->value);
        }

}
```

```
/*
=================================================
        DegradeTransferProbabilities
  degrades transfer probabilities of individuals
              in the Mating Set
=================================================
*/

void DegradeTransferProbabilities(int Ind_Center, int SizeOfArray, CPopulation *ptoArray)
{
        for (int k=0; (k<SizeOfArray && k!=Ind_Center); k++)
        {
                if (AreNeighbours (Ind_Center, k, ptoArray))
                        (ptoArray + k)->TransferProbability =
                        float ((ptoArray + k)->TransferProbability * DegradingFactor);
        }
```

```
}
```

```
/*
=====================================================
       Select_RemStochSampWithoutReplUsingElitism()
   Selects Strings for reproduction based on a mixture of
 the Reminder Stochastic Sampling Without Replacement
             and a novel Elitism scheme
=====================================================
*/
void Select_RemStochSampWithoutReplUsingElitism(float TotalFitness)
{
 int PopIndex = ۰;

 /* Calculating Expected Number of individuals to be selected*/
 for (int i=۰; i<POPULATION_SIZE; i++)
       ExpectedNumber[i] = POPULATION_SIZE *
     ((ptoCPool + i)->SharedFitness / TotalFitness);

 /* Selection based on the Integer part of the EcpectedNumber */
 for (int k=۰; k<POPULATION_SIZE; k++)
 {
for (int j=۰; j<int(ExpectedNumber[k]); j++)
{
//selected[PopIndex] = k;
selected[PopIndex] = (ptoCPool + k);
PopIndex++;
}
 }

 /* Selection based on the Fractional part of the EcpectedNumber */

/* Initializing Transfer Probabilities */
for (int kk=۰; (kk<EliteSet_Size &&
((ptoCEliteSet+kk)->NondominatedFront==۱)); kk++)
{
(ptoCEliteSet + kk)->TransferProbability = float
(InitialTransferProbability);
}

 while (PopIndex < POPULATION_SIZE)
 {
for (int jj=۰; (jj<POPULATION_SIZE && PopIndex<POPULATION_SIZE); jj++)
{



if (flip(ExpectedNumber[jj] - int(ExpectedNumber[jj])))
{
if (flip(P_Elitism)) // Apply Elitism
{
int Candidate = SelectFromEliteSet();
selected[PopIndex] = (ptoCEliteSet + Candidate);
}

else
{
selected[PopIndex] = (ptoCPool + jj);
}

PopIndex++;
}
} // 'for' loop ends here
 }
}
```

```
/*
=======================================================
```

```
            SelectFromEliteSet()
selects an individual for elitism from the Reference Set
==========================================================
*/
int SelectFromEliteSet()
{
int SelectedInd = -۱;
int i;

while (SelectedInd == -۱)
{
  for (i=۰; (i<EliteSet_Size &&
  ((ptoCEliteSet+i)->NondominatedFront==۱) &&
  (SelectedInd == -۱)); i++)
  {
if (flip ((ptoCEliteSet+i)->TransferProbability))
SelectedInd = i;
  }
}

DegradeTransferProbabilities(SelectedInd, EliteSet_Size, ptoCEliteSet);

return (SelectedInd);

}
```

```
/*
============================================
              crossover
         Swaps ۲ sub_Strings
============================================
*/
//void crossover(int parent۱, int parent۲, int child۱, int child۲)
void crossover(CPopulation* parent۱, CPopulation* parent۲, int child۱, int child۲)

{
int i, site;

if (flip(PCROSS))
site = random(CHROM_LENGTH);



 else
   site = CHROM_LENGTH - ۱;

    for (i=۰; i<CHROM_LENGTH; i++)
{
       if((i <= site) || (site == ۰))
   {
   //CNewPool[child۱].String[i] = CPool[parent۱].String[i];
           //CNewPool[child۲].String[i] = CPool[parent۲].String[i];
   CNewPool[child۱].String[i] = (parent۱)->String[i];
           CNewPool[child۲].String[i] = (parent۲)->String[i];
   }else
   {
   //CNewPool[child۱].String[i] = CPool[parent۲].String[i];
           //CNewPool[child۲].String[i] = CPool[parent۱].String[i];
   CNewPool[child۱].String[i] = (parent۲)->String[i];
           CNewPool[child۲].String[i] = (parent۱)->String[i];
   }
```

```
}
}
```

```
/*
==========================================
                mutation
    Changes the values of String position
==========================================
*/
void mutation(void)
{
int i, j;

    for (i=٠; i<POPULATION_SIZE; i++)
{
       for (j=٠; j<CHROM_LENGTH; j++)
   if(flip(PMUT))
   (ptoCPool + i)->String[j] = ~CNewPool[i].String[j] & ٠x٠١;
   else
               (ptoCPool + i)->String[j] = CNewPool[i].String[j] & ٠x٠١;
}
}
```

```
/*
==========================================
           Inversion
       Performs Inversion Operator
==========================================
*/
void Inversion()
{
unsigned char CutSection[CHROM_LENGTH];
int A, B, Site_١, Site_٢;
for (int i=٠; i<POPULATION_SIZE; i++)
{
if (flip(P_Inversion))
{
          // selection of two cut points along the chromosome length
  do
  {





A = random(CHROM_LENGTH);
B = random(CHROM_LENGTH);
  }
  while (A == B);

  if (A < B)
  {
  Site_١ = A;
  Site_٢ = B;
  }
  else
  {
  Site_١ = B;
  Site_٢ = A;
  }


  /* copying the cut section elements into the
     temporary array CutSection */
```

```
  for (int j=Site_۱; j<=Site_۲; j++)
CutSection[j] = CNewPool[i].String[j];

  for (int k=۰; k<=(Site_۲ - Site_۱); k++)
  CNewPool[i].String[Site_۱ + k] = CutSection[Site_۲ - k];

  /*
  TempBit = CNewPool[i].String[site_۱];
  CNewPool[i].String[site_۱] = CNewPool[i].String[site_۲];
  CNewPool[i].String[site_۲] = TempBit;
  */
}
}
}
```

```
/*
==========================================================
UpdateNonDominatedFrontOfPreviousRun()
updates file of the found front of previous run
     and replaces all dominated solutions
==========================================================
*/
void UpdateNonDominatedFrontOfPreviousRun()
{

int i, j, k, t;
int NDSolutionsCurrentRun = ۰; // number of nondominated solutions found by the
base algorithm
int NDSolutionsPreviousRun = ۰; // number of nondominated solutions found by the
reference algorithm
int UpperLimitCurrentRun = ۰;
int UpperLimitPreviousRun = ۰;
char DominatedIndividual = 'N'; // stands for "None"
int DominatedSolutions_CurrentRun = ۰; // number of dominated solutions in the
front ۱ of the base algorithm
int DominatedSolutions_PreviousRun = ۰; // number of dominated solutions in the
front ۱ of the  algorithm
FILE* pFront; // to print out final aggregate front
for (i=۰; i<POPULATION_SIZE; i++)
{
if ((ptoCPool+i)->NondominatedFront == ۱)
{
*(ptoC_NDFrontCurrentRun + NDSolutionsCurrentRun) = *(ptoCPool +
i);









     //(ptoC_NDFrontCurrentRun + i)->SourceAlgorithm= ۱; // the base algorithm
(ptoC_NDFrontCurrentRun + i)->Nondominated = true ; // just for
initialization in the base algorithm
NDSolutionsCurrentRun++;
}
}
UpperLimitCurrentRun = NDSolutionsCurrentRun;
for (j=۰; j<(UpperLimitCurrentRun - ۱); j++)
{
```

```
k=(j+1);
while (k < UpperLimitCurrentRun)
{
if ((ptoC_NDFrontCurrentRun + j)->value == (ptoC_NDFrontCurrentRun
+ k)->value)
{
for (t=k; t<UpperLimitCurrentRun; t++)
*(ptoC_NDFrontCurrentRun + t) =
*(ptoC_NDFrontCurrentRun + (t+1));
// to pull all individuals one position up

NDSolutionsCurrentRun--;
UpperLimitCurrentRun--;
k--;

} // end of the 'if' loop
k++;
} // end of the 'k' loop

} // end of the 'j' loop
pFront = fopen (UpdatingFile, "r"); // opens output file of the previous run
NDSolutionsPreviousRun = ReadFrontOneOfPreviousRun(pFront);
fclose (pFront);
UpperLimitPreviousRun = NDSolutionsPreviousRun;
for (j=0; j<UpperLimitPreviousRun; j++)
{
k=(j+1);
while (k < MaxAntNDFrontSize)
{
if ((ptoC_NDFrontPreviousRun + j)->value ==
(ptoC_NDFrontPreviousRun + k)->value)
{
for (t=k; t<(MaxAntNDFrontSize - 1); t++)
*(ptoC_NDFrontPreviousRun + t) =
*(ptoC_NDFrontPreviousRun + (t+1));
// to pull all individuals one position up

NDSolutionsPreviousRun--;
UpperLimitPreviousRun--;
k--;

} // end of the 'if' loop
k++;
} // end of the 'k' loop

} // end of the 'j' loop

for (i=0; i<NDSolutionsCurrentRun; i++)
{
if ((ptoC_NDFrontCurrentRun + i)->Nondominated == true)
{
for (j=0; j<NDSolutionsPreviousRun; j++)
{
if ((ptoC_NDFrontPreviousRun + j)->Nondominated == true)
{
DominatedIndividual = DominatedCheck(i,
ptoC_NDFrontCurrentRun, j, ptoC_NDFrontPreviousRun);





if (DominatedIndividual == 'F') // First element is dominated
{
// if the element has not been marked as
dominated
if ((ptoC_NDFrontCurrentRun + i)-
>Nondominated == true)
{
```

```
(ptoC_NDFrontCurrentRun + i)-
>Nondominated = false;
//DominatedSolutions_CurrentRun++;
}
}
}

else if (DominatedIndividual == 'S') // Second
element is dominated
{
// if the element has not been marked as
dominated
if ((ptoC_NDFrontPreviousRun + j)-
>Nondominated == true)
{
(ptoC_NDFrontPreviousRun + j)-
>Nondominated = false;
//DominatedSolutions_PreviousRun++;
}
}
}
}
}
}

//RPI_CurrentRun = float (NDSolutionsCurrentRun - DominatedSolutions_CurrentRun) /
(NDSolutionsCurrentRun);                                        //

//RPI_PreviousRun = float (NDSolutionsPreviousRun -
DominatedSolutions_PreviousRun) /
                            (NDSolutionsPreviousRun);      //


/* nondominated individuals from the current run */
pFront = fopen (UpdatingFile, "w"); // replaces output file of the previous run
int RowNumber = ۰;

for (i=۰; i<NDSolutionsCurrentRun; i++)
{
if ((ptoC_NDFrontCurrentRun + i)->Nondominated == true)
{
RowNumber++;
fprintf (pFront, "\n[٪۳d]*", RowNumber);

fprintf (pFront, "\t٪۲۰.۰f\t", (ptoC_NDFrontCurrentRun + i)->value);


for (int j=۰; j<CHROM_LENGTH; j++)
fprintf (pFront, "%d", (ptoC_NDFrontCurrentRun + i)-
>OldString[j]); // string before crossover and mutation

fprintf(pFront,"\t\t٪۱.٤f  ٪۱.٤f  ٪۱.٤f  ٪۱.٤f\t",

 (ptoC_NDFrontCurrentRun + i)->fitnessF۱,

 (ptoC_NDFrontCurrentRun + i)->fitnessF۲,

 (ptoC_NDFrontCurrentRun + i)->fitnessF۳,

 (ptoC_NDFrontCurrentRun + i)->fitnessF٤);
}

}

/* nondominated individuals from the previous run */
for (k=۰; k<NDSolutionsPreviousRun; k++)
{
if ((ptoC_NDFrontPreviousRun + k)->Nondominated == true)
{
RowNumber++;
```

```
fprintf (pFront, "\n[%۳d]*", RowNumber);
fprintf (pFront, "\t%۲۰.۰f\t", (ptoC_NDFrontPreviousRun + k)-
>value);

for (int j=۰; j<CHROM_LENGTH; j++)
fprintf (pFront, "%d", (ptoC_NDFrontPreviousRun + k)-
>OldString[j]); // string before crossover and mutation

fprintf(pFront,"\t\t%٦.٤f  %٦.٤f  %٦.٤f  %٦.٤f\t",
 (ptoC_NDFrontPreviousRun + k)->fitnessF۱,
 (ptoC_NDFrontPreviousRun + k)->fitnessF۲,
 (ptoC_NDFrontPreviousRun + k)->fitnessF۳,
 (ptoC_NDFrontPreviousRun + k)->fitnessF٤);

}
}
fclose (pFront);
}
```

پیوست ۳

مسائل مورد استفاده درارزیابي
الگوریتم XGA

P A R T S

MACHINES	۱	۳	۸	۹	۱۹	۶	۷	۱۰	۱۱	۱۷	۲	۵	۱۴	۲۷	۱۶	۱۸	۲۰	۲۲	۲۶	۱۲	۲۴	۲۱	۲۳	۲۹	۳۰	۱۳	۱۵	۴	۲۵	۲۸
۳	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱					۱													۱		
۷	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱																				
۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱																				
۲	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱																				
۱۰	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱																				
۶											۱	۱	۱	۱	۱	۱	۱	۱	۱	۱										
۸											۱	۱	۱	۱	۱	۱	۱	۱	۱	۱										
۴											۱	۱	۱	۱	۱	۱	۱	۱	۱	۱						۱	۱			
۵											۱	۱	۱	۱	۱	۱	۱	۱	۱	۱										
۹											۱	۱	۱	۱	۱	۱	۱	۱	۱	۱										
۱۱	۱	۱																			۱	۱	۱	۱	۱	۱	۱	۱	۱	۱
۱۲																					۱	۱	۱	۱	۱	۱	۱	۱	۱	۱
۱۳																					۱	۱	۱	۱	۱	۱	۱	۱	۱	۱
۱۴																					۱	۱	۱	۱	۱	۱	۱	۱	۱	۱
۱۵																					۱	۱	۱	۱	۱	۱	۱	۱	۱	۱

شکل (۱). مسئله VN۹۲ [۱]

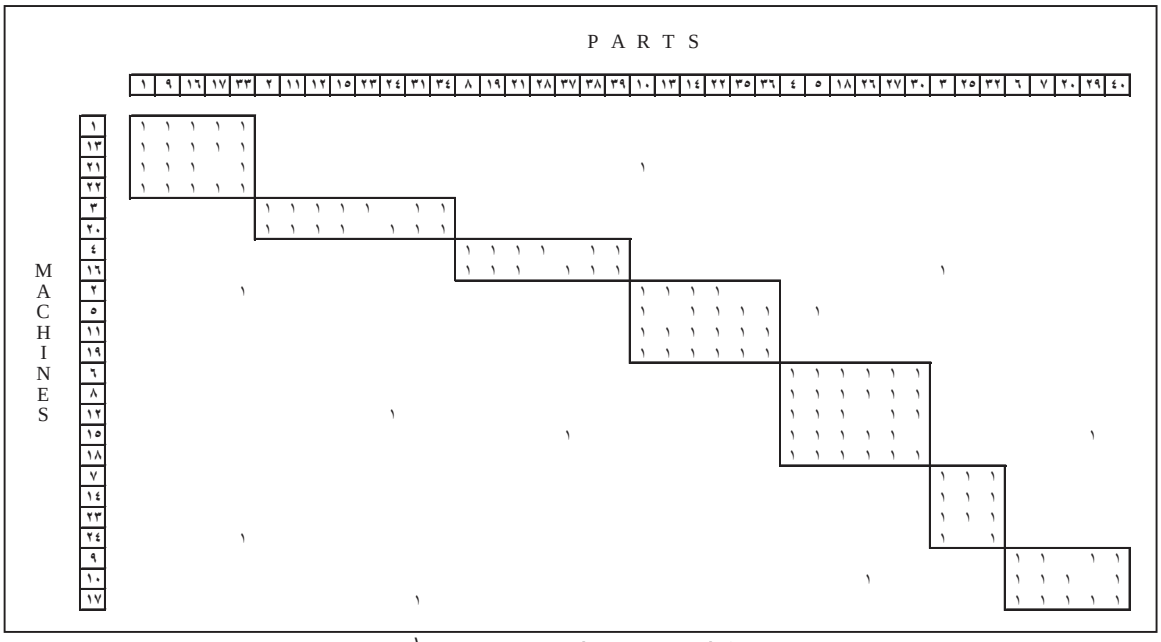
P A R T S

MACHINES	۲	۷	۱۰	۱۲	۱۳	۱۸	۲۴	۲۷	۳۱	۴	۶	۹	۱۱	۲۱	۲۸	۳۰	۳۲	۱۶	۳۳	۳۴	۳۵	۱	۳	۵	۱۵	۱۷	۲۰	۲۳	۲۵	۲۹	۸	۱۴	۱۹	۲۲	۲۶
۲	۱	۱	۱	۱	۱	۱	۱	۱	۱																										
۴	۱	۱		۱	۱		۱	۱																											
۱۳	۱			۱	۱		۱																												
۱۴	۱	۱	۱	۱	۱	۱	۱	۱	۱																										
۱۸	۱		۱	۱	۱	۱	۱		۱																										
۱۱										۱	۱	۱	۱	۱	۱	۱	۱				۱														
۱۲										۱	۱	۱	۱	۱					۱																
۱۵										۱	۱	۱	۱	۱	۱	۱																			
۱۶										۱	۱	۱	۱	۱	۱	۱	۱																		
۱۹										۱	۱	۱	۱	۱	۱	۱	۱																		
۵																		۱		۱											۱	۱			
۱																						۱	۱				۱	۱	۱						
۳									۱													۱	۱	۱	۱	۱				۱					
۷																						۱	۱	۱	۱	۱	۱	۱							
۸																						۱	۱	۱	۱	۱	۱	۱	۱	۱					
۱۷																						۱	۱	۱	۱	۱	۱	۱	۱	۱					
۶																		۱		۱											۱	۱	۱	۱	۱
۹																															۱	۱	۱	۱	۱
۱۰																		۱													۱	۱	۱	۱	۱
۲۰																												۱			۱	۱	۱		۱

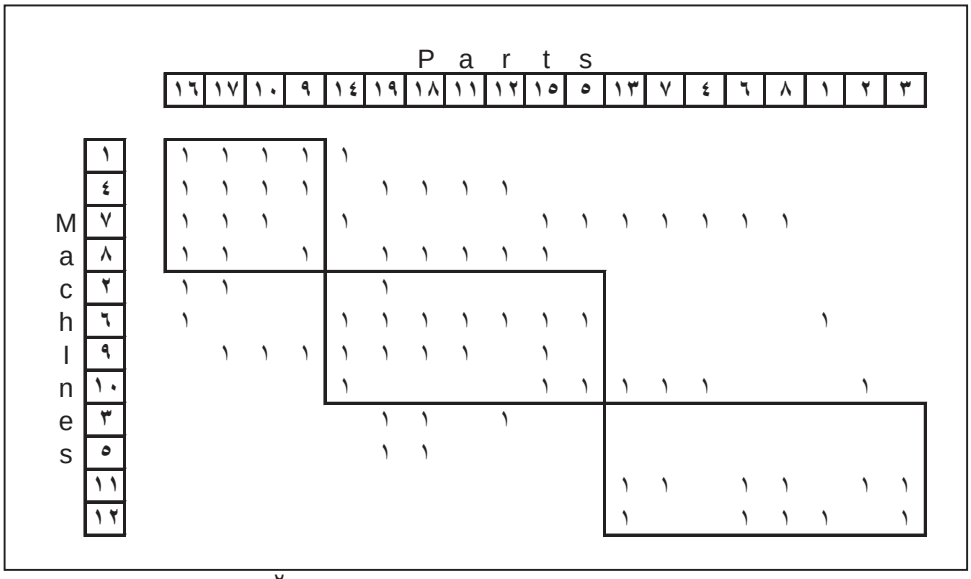
شکل (۲). مسئله Bur۶۹ [۲]

[۱] ونوگوپال و نارندران (۱۹۹۲)

[۲] باربیچ (۱۹۶۹)



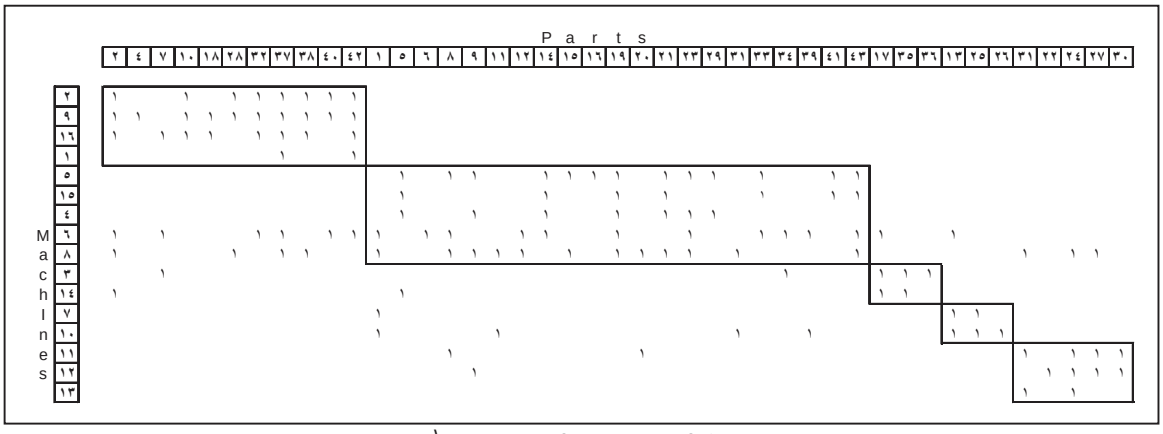
شکل (۳). مسئله KLA۹۷ [۱]



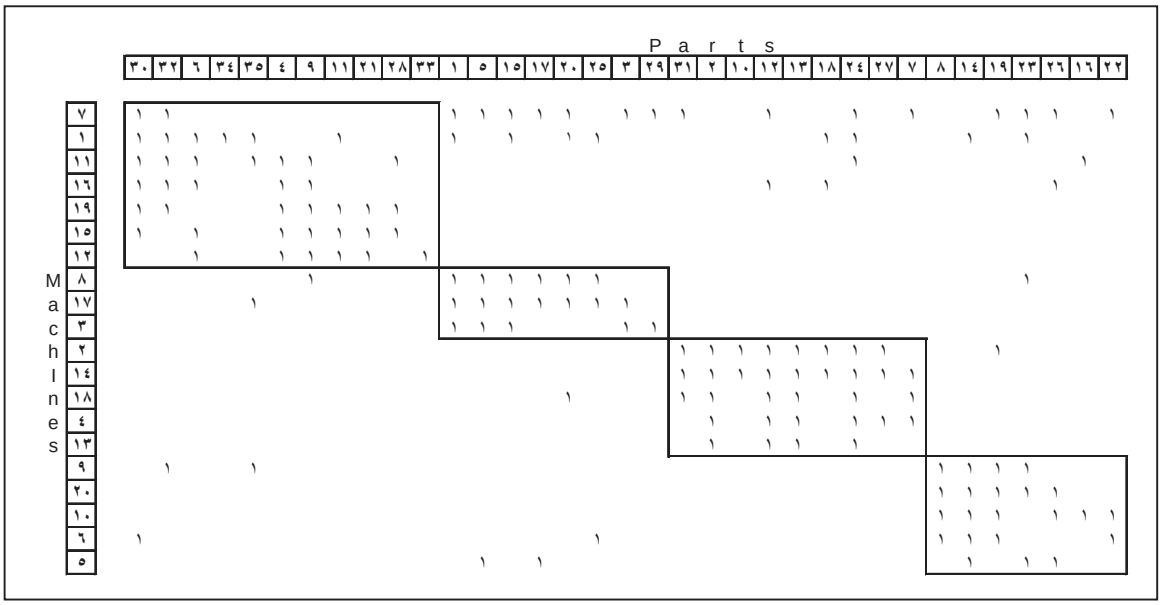
شکل (٤). مسئله ACGV۹۱-P۲ [۲]

[۱] کازروني و همکاران (۱۹۹۷)

[۲] اسکین و همکاران (۱۹۹۱)



شکل (۵). مسئله Sei۸۹ [۱]



شکل (٦). مسئله BC۹۱ [۲]

[۱] سیف الدیني (۱۹۸۹)
[۲] بو و چنگ (۱۹۹۱)

پیوست ۴

کپي مقاله:

A review of the modern approaches to
multi-criteria cell design

INT. J. PROD. RES., 2000, VOL. 38, NO. 5, 1201–1218

A review of the modern approaches to multi-criteria cell design

S. A. MANSOURI†, S. M. MOATTAR HUSSEINI† and S. T. NEWMAN‡*

The purpose of this paper is to provide a review and comparison of the approaches to multi-criteria decision-making (MCDM) in the design of manufacturing cells. A brief description of existing classifications is provided, together with an overview on the MCDM. Selected papers are reviewed and a structured scheme is outlined which allows comparison of inputs, criteria, solution approaches and outputs across selected models. Finally the models are discussed and directions for new avenues of work are identified.

1. Introduction

Cellular manufacturing (CM) is an important application of group technology (GT) in which sets (families) of parts are produced in manufacturing cells or a group of various machines, which are physically close together and can entirely process a family of parts. The identification of part families and machine groups in the design of cellular manufacturing systems is commonly referred to as cell design/formation. There have been many efforts towards the design of manufacturing cells based on the selection of part families and machine groups, considering only a single criterion such as minimising inter-cell movement of parts. There has been pressure on the manufacturing industries in the world market competition to improve their performances with regard to such measures as shorter delivery lead-times, wider range of products, shorter set-up times, and of course lower prices. These pressures provide a number of conflicting criteria on which performance is evaluated. Thus the design of the manufacturing area is critical to the efficient performance of the business.

From a system designer's point of view it is very desirable to achieve an optimal solution with respect to all the criteria considered individually by researchers, but this is impossible because of the conflicts between various criteria. For example, minimising inter-cell movement of parts by means of machine duplication/part subcontracting deteriorates the objectives of maximising machine utilisation and minimising total cost. In fact, based on Keen (1977), optimisation in the traditional mathematical sense is impossible if multiple criteria are involved. Problems such as trade-off analysis between machine duplication, part subcontracting, and intercellular material handling costs should be addressed and presented as a decision model to the decision-maker (Offodile *et al.* 1994).

The purpose of this paper is to review those papers, which consider the cell design problem as a multi criteria decision-making problem. The initial section of the paper

Revision received August 1999.

† Department of Industrial Engineering, Amirkabir University of Technology, Tehran 15914, Iran.

‡ Department of Manufacturing Engineering, Loughborough University, Leicestershire, LE11 3TU, UK.

* To whom correspondence should be addressed. e-mail: S.T.Newman@lboro.ac.uk

International Journal of Production Research ISSN 0020–7543 print/ISSN 1366–588X online © 2000 Taylor & Francis Ltd
http://www.tandf.co.uk/journals/tf/00207543.html

provides a brief description of the previous reviews on cell design models. The major part of the paper outlines an overview on the multi criteria decision making, together with the classification scheme used in the current survey. The final section provides a comparative discussion on the approaches used to deal with various criteria in the selected papers, a discussion on the gaps in the research area and future trends of new works.

2. A review on the previous work

Many models and solution approaches have been developed to deal with the problem of manufacturing cell design/formation since 1970s. The most common criteria considered for the design of manufacturing cells and a selection of research from the current decade is given in table 1.

There have also been some comprehensive review papers with different aims and viewpoints. King and Nakornchai (1982) provide a comprehensive review of the various approaches to the cell formation problem by the time. They classify all the models into four subdivisions: similarity coefficient, set theoretic, evaluative, and other analytical methods. Mosier and Taube (1985) partition the literature under four subtitles: part identification, grouping, scheduling in the GT shop, and GT implementation. Wemmerlöv and Hyer (1986) review more than 70 papers and categorise them into four groups. Their classification scheme divides all the methods into two major groups based on the main data for grouping as either part attributes or routings. The latter branch is further classified into three divisions, i.e. approaches that identify firstly the machine groups and then part families, approaches that identify firstly the part families and then machine groups, and the approaches that identify part families and machine groups simultaneously. Chu (1989) provides a comprehensive bibliography of cellular manufacturing and partitions the literature into design-oriented and production-oriented approaches. The latter group is further partitioned into array-based, hierarchical, non-hierarchical, mathematical, graph theoretic, and heuristic approaches. Offodile *et al.* (1994) employ a taxonomic framework and divide all the methods for identifying machine-part families into three taxonomic frameworks of: visual methods, parts coding analysis, and production flow analysis. The models in the latter class up to 1991 are compared based on their solution approach, decision variables, objectives and constraints.

Criterion	Selection references
minimising inter-cell movements	Klincewics and Rajan (1994), Joines *et al.* (1996), Sofianopoulou (1997), Cheng *et al.* (1996)
maximising parts and/or machines similarities (or minimising dissimilarities)	Kao and Moon (1991), Kaparthi and Suresh (1991), Chu (1993), Boctor (1991), Chen and Srivasata (1994), Askin *et al.* (1991), Lee and Carcia-Diaz (1993), Kusiak *et al.* (1993)
obtaining the block diagonal form of the part-machine incidence matrix	Kaparthi and Suresh (1992), Kaparthi *et al.* (1993), Suresh and Kaparthi (1994), Mukhopadhyay *et al.* (1994), Chen and Irani (1993)
minimising cell load unbalances	Venugopal and Narendran (1992 b)
minimising number of exceptional elements	Song and Hitomi (1992), Amirahmadi and Choobineh (1996)

Table 1. Some single criterion models for cell design in the 1990s.

None of the above mentioned reviews considers the number of criteria (single or multiple) as a measure for classification. The aim of this paper is to establish this measure as a major classifier on the cell design models and to present a review on the multi criteria models. It is not the intention of this paper to review single criterion models for cell design; readers are referred to King and Nakornchai (1982), Mosier and Taube (1985), Wemmerlöv and Hyer (1986), Chu (1989), and Offodile *et al.* (1994).

3. An overview on the multi-criteria decision-making

Criteria are measures, rules and standards that guide decision-making. As selecting or formulating different attributes, objectives or goals conducts decision-making, all three categories can be referred to as criteria (Zeleny 1982). The multi-criteria decision-making (MCDM) problems based on the above definitions are categorised by Zeleny (1982) as: multi-attribute decision-making (MADM), multi-objective decision-making (MODM) and goal programming (GP). The manufacturing cell formation problem, taking multiple criteria into consideration, becomes compatible with the MODM and GP problems. For a more detailed comparison of MADM and MODM problems, readers are referred to Hwang and Yoon (1981).

3.1. *Multi-objective decision-making*

In its most general form, the MODM problem can be stated mathematically as follows:

$$\begin{aligned} &\max\,[f_1(x),\ldots,f_k(x)] \quad \text{subject to} \\ &g_i(x) \le 0, \quad \text{for } i = 1,\ldots,m, \qquad x \in X \end{aligned} \tag{1}$$

where X is some subset of R^n, f_1 are objective functions and the g_i are constraints. This problem includes n decision variables, m constraints and k objectives. It is possible that some or all of the functions are nonlinear. The problem (1) is also known as the vector maximum problem or VMP. A solution to the problem (1) is defined as non-dominated if no other solution exists which could improve one or more objectives without detriment to some other objective. A necessary and sufficient condition for a solution to be efficient is described by Hartley (1983) as follows: a feasible $x \in X$ is efficient if and only if there is no feasible to satisfy $y \in X$ to satisfy $f(y) >= f(x)$, where f is the vector-valued function whose i'th component at x is $f_i(x)$ and $=, >$ denote, respectively, weak and strict component-wise inequality whilst $>=$ denotes '$=$ but not $=$'. Two well-known approaches to solve the VMP are the constraint method and the weighting method. In the constraint method, one of the objectives is selected for optimisation and the other objectives are added to the set of constraints in the form of inequalities with a minimum acceptable level. In the constraint method, problem (1) is reformulated as follows:

$$\begin{aligned} &\max f_j(x) \quad \text{subject to} \\ &g_i(x) \le 0, \quad \text{for } i = 1,\ldots,m \\ &f_i(x) \ge \varepsilon_l, \quad \text{for } l = 1,\ldots,k;\, l \ne j,\ x \in X \end{aligned} \tag{2}$$

where ε_l denotes the minimum acceptable level for the lth objective.

In the weighting method, a single objective function, i.e. the weighted sum of the individual objectives is defined, and in this manner, the multi-objective problem is changed to a single objective problem as follows:

$$\begin{aligned} &\max \sum_{l=1}^{k} w_l \cdot f_l(x) \quad \text{subject to} \\ &g_i(x) \leq 0, \quad \text{for } i = 1, \ldots, m, \; x \in X \end{aligned} \tag{3}$$

The weights assigned to the objectives are usually normalised, i.e.

$$\sum_{l=1}^{k} w_l = 1.$$

A major drawback of the above methods is the dependency of the solution to the values of ε_l and w_i which are determined subjectively by the decision-maker *a priori*. For a comprehensive study of the approaches to the MODM problem, readers are referred to Hwang and Masud (1978) and Szidarovszky *et al.* (1986).

3.2. *Goal programming*

The notion of goal programming (GP) is the attempt to minimise the set of deviations from prespecified multiple goals, which are considered simultaneously but are weighted according to their relative importance. A general GP problem can be formulated as follows:

$$\begin{aligned} &\min \sum_{i=1}^{m} p_i(d_i^+ + d_i^-) \quad \text{subject to} \\ &\sum_{j=1}^{n} c_{ij} \cdot x_j + d_i^+ - d_i^- = b_i \quad \text{for } i = 1, \ldots, m \end{aligned} \tag{4}$$

where x_j are n decision variables, d_j^- and d_i^+ denote, respectively, negative and positive deviations from the ith goal (or slack and surplus variables in case an equation represents an ordinary constraint), b_i are m goals or rigid constraining values, and c_{ij} are technological coefficients. p_i in the objective function stand for pre-emptive weights or priority weights determining the hierarchy of goals. Goals of higher priority levels are satisfied first, and only then are the lower priority goals considered.

The model (4) could be handled as a single objective minimisation problem by means of the traditional techniques of linear, non-linear, integer, or mixed integer programming where appropriate. For a detailed discussion on various approaches to solve the GP problem, readers are referred to Hillier and Lieberman (1986) and Zeleny (1982).

4. Review of papers

The design of manufacturing cells with respect to multiple criteria has been an attractive research topic since 1990. This section presents a review on the main features of the models developed in this field. The main criterion for the selection of papers for this study has been the consideration of at least two criteria simultaneously in the solution approach of the model. Therefore such research as Vannelli and Hall (1993) and Hadley (1996), which separate machine group/part family for-

mation in two stages that consider one independent objective at each stage, are not included.

Sankaran (1990), to consider multiple goals in the cell formation procedure, primarily solves a single objective model whose objective function is the sum of five distinct cost functions. The optimal solution of the single objective model is then broken down into two cost aspiration levels, i.e. operating cost and capital investment cost. These two optimal costs along with five other goals are then combined in an integer linear goal programming model. The set of the goals considered by the author includes: minimum similarity of parts based on their needed machines and tools (two goals), available machining capacity, minimum and maximum number of total parts movement (two goals), the optimal capital investment on machines, and the optimal operating cost.

Wei and Gaither (1990) develop a four objective cell formation model to minimise the bottleneck cost, maximise the average cell utilisation, minimise intra-cell load imbalances, and minimise inter-cell load imbalances. The bottleneck cost may be either the cost difference between making a bottleneck part inside and outside the cellular system, or the cost incurred by transporting the bottleneck part between cells. The authors develop a 0–1 programming model for small problems and a heuristic to solve both small and large problems. Both the optimal model and the heuristic seek to maximise a weighted additive utility function comprised of the above four objectives.

Shafer and Rogers (1991) apply goal programming in three unique situations: setting up an entirely new system and purchasing all new equipment, reorganising the system using only existing equipment, and reorganising the system using existing and some new equipment. The latter model, which is more compatible with the rest of the cell design models, is selected for detailed comparison in the next section. The criteria considered by the authors include: minimising set-up times (through parts sequencing), minimising intercellular movements, minimising the investment in new equipment, and maintaining an acceptable machine utilisation level. The proposed goal programming models combine the p-median (for identifying part families) and the travelling salesman problem (to determine the optimal sequence of parts). The authors then propose a heuristic to solve realistically sized problems.

Shafer *et al.* (1992) present a mathematical programming model for dealing with exceptional elements. They develop an initial solution using any of the cell formation procedures. Exceptional elements are then primarily eliminated by changing the design or process plans of the parts. Further elimination of the exceptional elements is carried out using part subcontracting or machine duplication through an optimisation model. The authors include three cost factors in the form of a single minimisation objective function, i.e. cost associated with intercellular transfer, machine duplication, and part subcontracting. In fact they use a weighting approach to unify three different objectives with equal weights.

Venugopal and Narendran (1992 a) propose a bi-criteria mathematical model for the machine-component grouping problem. The authors consider two minimisation objectives: minimise the volume of inter-cell moves, and minimise the total within cell load variation. Their solution approach is based on finding satisfactory or compromise solutions by means of genetic algorithms.

Dahel and Smith (1993) develop a 0–1 integer model for minimising inter-cell moves and a multi-objective model to form cells which are both flexible and have minimum interactions based on the results of the first model. They measure flexibility

of a cell (routeing flexibility) based on the number of different parts which could be handled by the cell. A greater degree of routing flexibility can be achieved by assigning a greater variety of machines to a cell, that in turn increases workflow among the cells and thus deteriorates the objective of minimising inter-cell moves. The authors use the constraint method (Cohon 1978) to obtain non-dominated solutions for the multi objective model.

Logendran (1993) develops a model to minimise total inter-cell and intra-cell movement of parts and to maximise cell machine utilisation. These objectives are unified through the weighting approach in the form of a single objective. The original model is formulated as a quadratic binary programming model and then converted into a linear binary programming model.

Min and Shin (1993) attempt to form both machine cells and human cells simultaneously by means of a mixed integer goal programming model. The goals are concerned with the level of parts similarity, available machine processing times, machine capabilities/operator skill matching, and the difference between the wages of the cells operators and the rest of the operators. A heuristic is then developed for machine cell formation and operators allocation in two subsequent stages.

Sankaran and Kasilingam (1993) develop an integer programming model to determine cell membership in a GT-based flexible manufacturing system along with cell size and capacity selection. The developed objective function can be viewed as the sum of the processing costs, the spatial cost, the annual amortisation costs, and the inter-cell and intra-cell movement costs. In other words, the weighting approach is used to deal with multiple objectives. The authors then present a heuristic to solve the model.

Gupta *et al.* (1995) develop a minimisation model that takes into consideration the weighted sum of both inter-cell and intra-cell moves. The authors use a genetic algorithm for solving the model. There is attention on the minimum acceptable level of machine utilisation in the procedure of part assignment to the cells.

Liang and Taboun (1995) in an attempt to achieve a compromise solution between flow-line efficiency and job-shop flexibility develop a bi-criterion nonlinear integer programming model. The objectives of the model are to maximise system flexibility (measured based on the number of part types accommodated into the focused cells), and to maximise system efficiency (measured based on the degree of part similarities). The authors assume that no inter-cell movement is allowed and all exceptional parts should be sub-contracted. They use the weighting approach to unify the two objectives in the form of a single objective. A heuristic is then proposed which consists of two phases, i.e. generation of approximate efficient (or non-dominated) solutions, and calculation of the best solution based on the approximate efficient solution. The constraint method is used to generate approximate efficient solutions.

Suresh *et al.* (1995) employ a three-phase hierarchical approach to cell formation. In phase I of the solution approach, a neural network clustering technique is used to identify potential part families and their associated machine groups. A mixed integer goal programming formulation is then used in phase II to assign individual machines to a specified number of cells. The model of phase II attempts to satisfy the conflicting goals of: maximising the cell independence, minimising the purchase of new equipment and maximising the routing flexibility. Phase III aims to minimise inter-cell traffic further for families that may still have to be processed in more than one cell by means of a 0–1 programming model.

Akturk and Balkose (1996), by means of a coding scheme which includes both design and manufacturing attributes of parts, calculate the similarity and dissimilarity of parts and makes use of them in a six objective model. The objectives are concerned with minimising: the dissimilarity of parts based on the design and manufacturing attributes, the dissimilarities based on the operation sequences, the total machine investment cost, the sum of the workload variability in each cell, the workload variability of different cells, and the number of skipping which refers to the number of machines a part skips in its operation sequence. The authors suggest a multi-objective cluster analysis heuristic to deal with these objectives simultaneously. The analytic hierarchy process (AHP) is employed to determine priority of the objectives in order to unify them.

Boctor (1996) develops a number of mixed integer models, of which model P1 is selected as a multi-objective model for this study. The objective function of the model P1 is composed of two cost terms, i.e. machine duplication cost and inter-cell movement cost, which should be totally minimised. These two cost functions do not move in the same direction in such a way that decreases the second term, i.e. the cost of inter-cellular movements is achieved partly through machine duplication that in turn increases the first cost term. The author unified these two conflicting criteria in the form of a single objective function through a weighting approach with equal weights assigned to each criterion. Simulated annealing is then used to obtain good solutions for the optimisation model.

Gupta *et al.* (1996) present three models for the machine cell-part grouping problem. Their model 3 is a bi-criteria model that considers simultaneously the minimisation of total weighted inter-cell and intra-cell movements, and the minimisation of the cell load variations. The authors employ a genetic algorithm to provide the decision-maker with a set of satisfactory solutions.

Ho and Moodie (1996) assume flexible routeing for parts and have developed a two-stage solution approach. Part families are formed in stage 1 based on the similarities in their operation sequences by means of a heuristic procedure. Machines in stage 2 are allocated to the part families through a mixed integer programming model. The objective function of this stage is composed of three cost functions: operation cost, machine duplication cost, and a penalty cost for those operations that need to be performed in cells other than the ones that they have been assigned. The latter term could be considered as another definition for the inter-cell movement cost. The authors employ, in stage 2, the weighting approach to unify the three cost criteria.

Rajamani *et al.* (1996) develop a mixed integer programming model with the assumption of flexible process plans for parts. The objective function of the model is to minimise the sum of investment, process and material handling costs as a weighted sum of the three different cost functions. To solve the relaxed linear model the authors make use of a column generation scheme and the branch and bound technique.

Lee and Chen (1997) employ a weighting approach to combine two criteria, i.e. minimising inter-cell movement of parts, and maximising workload balance among duplicated machines. Their solution methodology is a three-phase approach, which determines machine cells and part families and allows for machine duplication where necessary. At the beginning, an estimation procedure is developed to determine workload balances for duplicated machines. In the second phase, machine cells and part families are constructed by means of a heuristic algorithm. Finally, a

heuristic procedure is employed to improve the solution quality of the formation result.

Su and Hsu (1998) consider three objectives in their model. The objectives are: minimising the total cost of inter-cell transportation, intra-cell transportation and machine investment; minimising intra-cell machine (in cell) load unbalance; and minimising inter-cell machine (in plant) load unbalance. The authors then unify these objectives through weighting, and solve the model by means of parallel simulated annealing. The authors make use of the crossover and mutation functions of a genetic algorithm to function as the generation mechanism of simulated annealing in order to cope with its main drawback, i.e. high execution time.

5. Comparison of the models and research direction

The comparison of the multi-criteria cell design models is carried out based on their inputs, criteria, solution approach and outputs. It should be noted that the classification scheme used in this study is partially influenced by the work of Offodile *et al.* (1994).

5.1. *Comparison based on the inputs*

The input data is primarily classified based on the source of the data. It is divided into four categories: parts data, machines data, constraints, and general data. The data related to the parts and machines are further classified based on the type of data such as: quantitative data and cost data. Figure 1 shows the classification of these inputs.

The input data of the previously discussed models are illustrated and compared in table 2. In this table the rows represent various inputs considered in the multi-criteria models. The most common input data of the models are required machines (by parts), process times, production capacity of machines, maximum cell size, and parts demand. It could be said that a base cell design model must contain at least such a set of input data. Other relatively common input sets include the number of available machines, the predetermined number of cells, the fixed process flow of parts, the acquisition cost of machines, the inter-cell transportation cost, and the minimum cell size. Fixed process flow, as it could be traced in table 2, is becoming

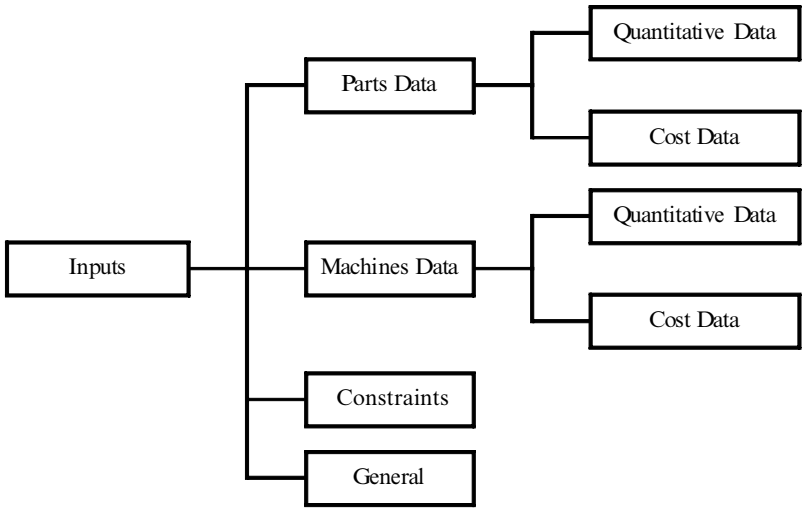


Figure 1. Classification of the inputs.

	Inputs	Sankaran (1990)	Wei and Gaither (1990)	Shafter and Rogers (1991)	Shafer *et al.* (1992)	Venugopal and Narendran (1992)	Dahel and Smith (1993)	Logendran (1993)	Min and Shin (1993)	Sankaran and Kasilingam (1993)	Gupta *et al.* (1995)	Liang and Taboun (1995)	Suresh *et al.* (1993)	Akturk and Balkose (1996)	Boctor (1996)	Gupta *et al.* (1996)	Ho and Moodie (1996)	Rajamani *et al.* (1996)	Lee and Chen (1997)	Su and Hsu (1998)
Parts data																				
Quantitative	Demand	*	*	*	*	*	*			*			*	*	*	*	*	*	*	
	Required machines	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
	Required tools	*																		
	Fixed process flow			*			*	*		*	*		*	*						*
	Alternative process flows																*	*		
	Alternative machines for perations												*	*						
	Process times	*	*	*	*	*	*	*	*	*		*	*	*	*	*	*	*	*	*
	Set-up times						*					*							*	*
	Batch size			*		*						*							*	
	Pallet capacity																		*	
	Part coding												*							
Cost	Direct prod. cost	*								*							*	*		
	Tool usage cost	*																		
	Operator wage								*											
	Subcontract cost.				*															
	Intercell transport. cost				*					*					*		*	*		*
	Intracell transport. cost	*								*										*
	Set-up cost									*										
	Space cost									*										
Machine data	Number of each machine			*			*	*		*	*	*	*		*	*	*			
	Production capacity	*	*	*	*	*	*	*	*		*	*		*	*	*	*	*	*	*
	Available tools	*																		
	Space requirements									*										
	Acquisition cost	*		*	*									*	*		*	*		
	Operation cost	*												*			*			
Constraints	Fixed number of cells	*	*	*		*	*	*						*	*	*		*		
	Interval of cells number										*						*			
	Max. utilisation													*						
	Min. utilisation										*		*	*						
	Max. cell size	*	*	*			*	*	*			*	*	*	*	*	*	*	*	*
	Min. cell size			*			*						*		*	*				
	Max. cell output																*			
	Operator limitations								*											
	Funds available			*																
	Space available									*										
General	Operator skills								*											
	Arrangement of machines in cells																			*
	Arrangements of cells in plant																*			*

Table 2. Comparison of the models based on their inputs

less important as a major input. That may be due to the increasing importance of flexibility that provides manufacturing system designers with more alternatives in process route of the parts. Such inputs as set-up times, the batch size, the direct production cost, the intra-cell transportation cost, and the minimum acceptable level of utilisation have been among the less common inputs. There have also been some inputs which are used in one single model only, e.g. the required tools for parts, the subcontracting cost, the space requirement of machines, and the operator limitations.

5.1.1. *Future work direction: input data*

There is a need to consider some new sets of input data in the cell design process to incorporate some of the real aspects of the current industrial environment. Obviously the most common inputs as identified above need to be within all models. Flexibility of manufacturing is becoming a significant importance in modern manufacturing, so it would be essential to incorporate such an important factor in the set of input data by quantifying alternative process routes and flexible machining processes. It is also important to consider the operation cost of machines to maximise utilisation of the machines for higher efficiency. The stochastic nature of demand is another important feature of the manufacturing environment that should be considered in the design of the manufacturing facility.

Standardisation of the input/output data sets could be another direction for future work. The models should provide various levels of inputs, with core inputs going to a master set of outputs, followed by optional inputs which allow secondary outputs to be provided, and so on. To this end, further research should be done in order to facilitate the definition of the minimum set of input data to achieve a specific set of output. Some field studies in various industries might be useful in identifying these sorts of minimum input/output data sets.

5.2. *Comparison based on the criteria*

The criteria used in the cell design have been classified by the authors under objectives and goals; objectives are classified by the authors, primarily based on their orientation, as cost-oriented, performance-oriented. Cost-oriented objectives are in the form of minimisation. Performance-oriented objectives are further divided into minimisation and maximisation. The goal programming models aim to minimise the deviation from the predetermined goals. The structure of this classification is shown in figure 2.

Table 3 gives a comparison of the models based on these criteria. There has been a vast diversity among the criteria considered by researchers. Minimising the machine duplication cost, minimising the inter-cell transportation cost, minimising the number of inter-cell movement of parts, and minimising the cell load unbalance of machines, have been the most popular criteria. In a lower level of importance, such criteria as minimising the intra-cell transportation cost and maximising flexibility are also included. There is also a set of unique criteria, especially in the form of goals, such as minimising deviation from the level of matching between operator skills and machine capabilities, minimising deviation from a predetermined level of machine investment cost, and minimising the amount of skipping. In addition to the diversity of criteria, the models with the same set of criteria were very scarce. In fact only Ho and Moodie (1996) and Rajamani *et al.* (1996) select the same set of criteria

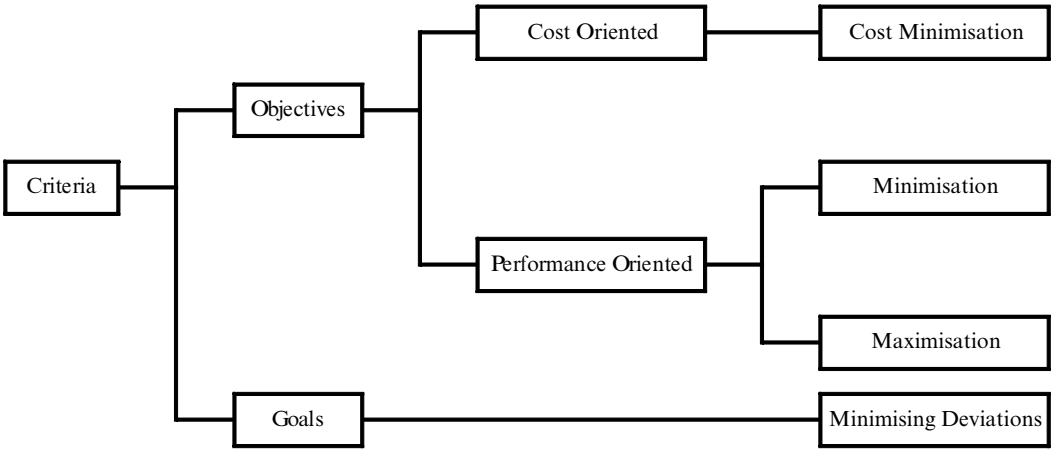


Figure 2. Classification of the criteria.

in their models. Thus it could be concluded that the majority of the multi-criteria cell design models are unique and cannot be compared with each other.

5.2.1. *Future work direction: criteria.*

A vast area of research on cell design could be to develop models with a different combination of criteria that have not been previously considered simultaneously. For example, models which consider the total cost minimisation and the flexibility maximisation, along with a combination of other objectives, have not been reported. Another possible trend would be to develop more efficient solution approaches for models with the same set of criteria as an existing model.

A different trend should be research on handling different criteria in the procedure of cell design. The majority of the models use the weighting approach to unify the multiple objectives in the form of a single objective. A sophisticated objective function is an outcome of this approach, besides the fact that the weighting approach in many cases deals with optimising a senseless objective function. A future possible trend for the research in this field is to develop more efficient solution tools enabling system designers to achieve good solutions in a reasonable processing time. To investigate on the more efficient ways for finding non-dominated solutions is a good area of research for those models that consider multiple criteria simultaneously.

5.3. *Comparison based on the solution approach*

The solution approaches are classified based on their solution techniques, into groups, which are presented in figure 3. Classification of the mathematical programming branch into constraint method, weighting method and goal programming is carried out based on models 2, 3 and 4 of section 3.1, respectively.

The solution approach of the models is compared in the top section of table 4. Mathematical programming has been the most common approach, where weighting method and goal programming have been applied most to the multi-criteria cell design. Goal programming applications were reported before 1995, and it seems that the approach lost its attraction in the second half of the 1990s. The constraint method has been employed in a single article. Due to the NP-complete nature of the problem, heuristics have been vastly employed to deal with real world problems. Genetic algorithms and simulated annealing have been the only reported search

	Criteria	Sankaran (1990)	Wei and Gaither (1990)	Shafer and Rogers (1991)	Shafer *et al.* (1992)	Venugopal and Narendran (1992)	Dahel and Smith (1993)	Logendran (1993)	Min and Shin (1993)	Sankaran and Kasilingam (1993)	Gupta *et al.* (1995)	Liang and Taboun (1995)	Suresh *et al.* (1995)	Akturk and Balkose (1995)	Boctor (1996)	Gupta *et al.* (1996)	Ho and Moodie (1996)	Rajamani *et al.* (1996)	Lee and Chen (1997)	Su and Hsu (1998)
Objectives																				
min.	Machine duplication				*									*	*		*	*		*
cost of	operation cost									*							*	*		
	Part subcontracting		*		*															
	Intercell transportation		*							*					*		*	*		*
	Intracell transportation				*					*										*
	Space usage									*										
min.	Duplicated machines												*							
amount of	Intercell movements					*	*	*			*		*			*				
	Intracell movements							*			*					*			*	
	In cell load unbalance		*			*								*		*				*
	In plant load unbalance		*											*					*	*
	Parts dissimilarity													*						
	Skippings													*						
max.	Flexibility						*					*	*							
amount of	Efficiency											*								
	Utilization		*					*												
	Cell independence												*							
min.	Parts similarity	*							*											
deviation	Machine available processing time	*							*											
from	Cell operator wages								*											
	Machine-operator skill matching								*											
	Minimum part movements	*																		
	Maximum part movements	*																		
	Machine investment cost	*																		
	Operating cost	*																		
	Set-up time			*																
	Utilisation			*																
	Available funds			*																
	Intercell moves			*																

Table 3. Comparison of the models based on their criteria

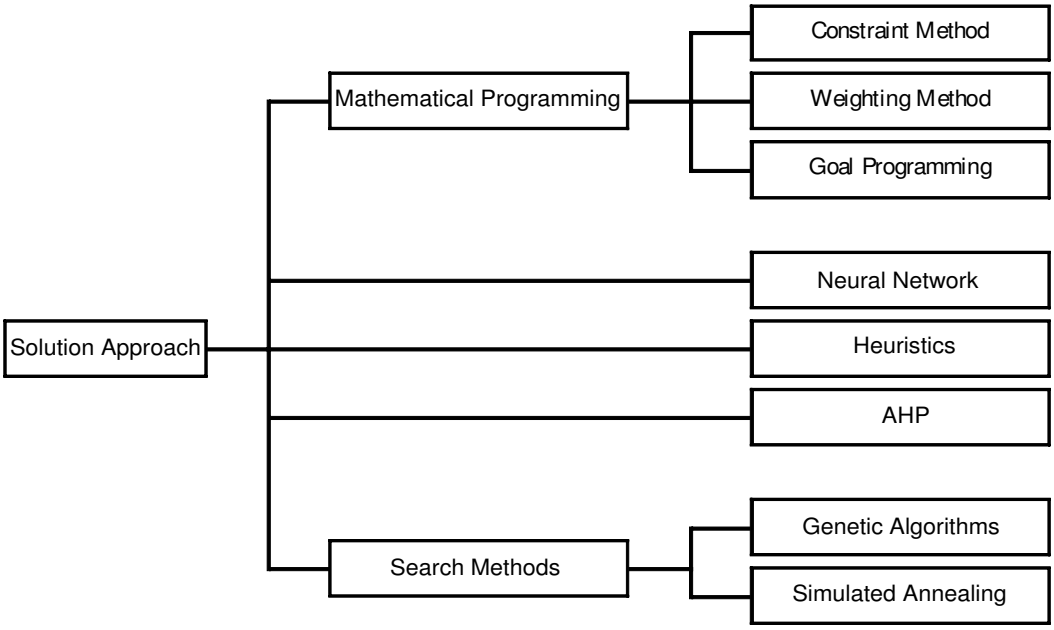


Figure 3. Classification of the solution approaches.

methods used to solve the models. Neural networks and the analytic hierarchy process (AHP) have been separately used as a part of the solution approach, along with other solution tools.

5.3.1. *Future work direction: solution approaches.*

The most common solution approaches in the mathematical programming category have been the weighting method and goal programming. These approaches at most are able to find a single non-dominated solution. If the solution is not good enough to satisfy the system designer, the model should be resolved with different parameters. More researches is necessary in applying other multi-objective optimisation techniques in such a way that a rich subset of true non-dominated solutions is identified, from which system designers could select. Employing other solution approaches like the Tabu search or fuzzy clustering are some other potential research areas. Trying to make use of the optimisation capabilities of genetic algorithms and neural networks, which have inherent capability of parallel processing in the multi-objective cell design, are some possible research areas which have not been researched significantly.

5.4. *Comparison based on the output*

Most of the models produce some information in addition to their major objectives. These additional outputs are classified based on their relation to the elements of grouping as part-related, machine-related, and general information. Figure 4 demonstrates the classification of the outputs.

The outputs of the models, in addition to those directly measured as criteria, are shown in the bottom section of table 4. Part families and machine groups are of course the main outputs of each cell design model. Duplicated machines, inter-cell movement, and machine utilisation have been among the common additional outputs. Duplicated machines had not been a major output in the first half of the decade, but since then have become a major output. There have also been some

		Sankaran (1990)	Wei and Gaither (1990)	Shafter and Rogers (1991)	Shafer *et al.* (1992)	Venugopal and Narendran (1992)	Dahel and Smith (1993)	Logendran (1993)	Min and Shin (1993)	Sankaran and Kasilingam (1993)	Gupta *et al.* (1995)	Liang and Taboun (1995)	Suresh *et al.* (1995)	Akturk and Balkose (1996)	Boctor (1996)	Gupta *et al.* (1996)	Ho and Moodie (1996)	Rajamani *et al.* (1996)	Lee and Chen (1997)	Su and Hsu (1993)
Solution approach	Constraint method						*													
	Weighting method		*		*			*		*		*			*		*	*		
	Goal programming	*		*					*				*							
	Neural network												*							
	Heuristics		*	*			*		*	*		*		*			*		*	
	AHP													*						
	Genetic algorithm					*					*					*				
	Simulated annealing														*					
Output (additional measures)	Part families	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
	Part related																			
	Subcontracted parts		*		*							*								
	Different process plans																	*		
	Intercell movements				*					*										*
	Intracell movements									*										*
Machine-related	Machine groups	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
	Duplicated machines				*									*	*		*	*	*	*
	In cell machine arrangement													*						*
	Utilisation										*					*				*
General	Parts sequencing			*																
	Human groups								*											
	In plant cell arrangement																			*

Table 4. Comparison of the models based on their solution approach and outputs.

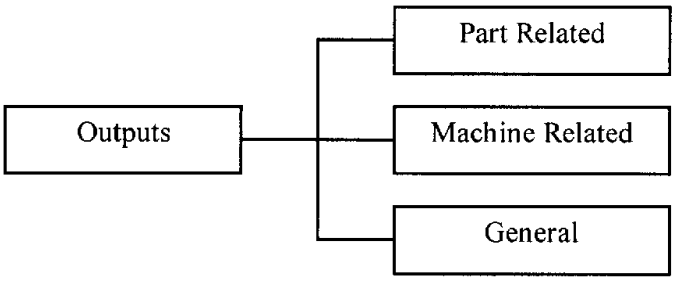


Figure 4. Classification of the outputs.

unique outputs such as different process plans, part sequencing and human groups which were considered for special cases and thus could not be regarded as a major output of a cell design model.

6. Conclusion

In this paper various approaches to the problem of multi-criteria cell design were investigated and reviewed. Previous reviews on cell design/formation were discussed and the need for this review was identified. A brief discussion on multi-criteria decision-making is given and multi-criteria cell design is categorised as either a multi-objective problem or a goal programming problem, along with an overview on these notions.

The reported papers on multi-criteria cell design were discussed and their modelling and solution approach were highlighted. Finally the models were compared, based on their inputs, criteria, solution approach and outputs. This comparison enables researchers to identify the research gaps in the literature.

Including some real aspects of the manufacturing environment (such as flexibility of manufacturing facilities as well as process routes, the stochastic nature of demand, and the standardisation of data sets) have been suggested for additional research work. Developing the new multi-objective cell design models with different sets of objectives is another possible research avenue. For example, models that consider cost minimisation and flexibility maximisation, along with maximising utilisation have not been reported. Such gaps in the modelling stage can easily be seen in table 3. Another possible trend could be work on the application of the other solution tools in the reported models. Minimising inter-cell part movement and maximising flexibility by means of multi-objective genetic algorithms, and minimising the sum of the cost of part subcontracting and inter-cell part movement by means of neural networks, are some examples of work to be investigated further in this field.

The majority of the works on multi-objective cell design unify the various objectives in the form of a single objective. The final result of such an approach is a compromise solution, whose non-dominance is not guaranteed. Pareto optimisation through simultaneous consideration of various objectives, and the ability to provide the decision-maker with a set of non-dominated solutions in a reasonable computation time, are other areas which need to be addressed.

References

AKTURK, M. S. and BALKOSE, H. O., 1996, Part-machine grouping using a multi-objective cluster analysis. *International Journal of Production Research,* **34**, 2299–2315.

AMIRAHMADI, F. and CHOOBINEH, F., 1996, Identifying the composition of a cellular manufacturing system. *International Journal of Production Research,* **34**, 2471–2488.

ASKIN, R. G., CRESSWELL, S. H., GOLDBERG, J. B. and VAKHARIA, A., 1991, A Hamiltonian path approach to reordering the part-machine matrix for cellular manufacturing. *International Journal of Production Research,* **29**, 1081–1100.

BOCTOR, F. F., 1991, A linear formulation of the machine-part cell formation problem. *International Journal of Production Research,* **29**, 343–356; 1996, The minimum cost, machine-part cell formation problem. *Ibid.,* **34**, 1045–1063.

CHEN, C. Y. and IRANI, S. A., 1993, Cluster first-sequence last heuristics for generating block diagonal forms for a machine-part matrix. *International Journal of Production Research,* **31**, 2623–2647.

CHEN, W. H. and SRIVASATA, B., 1994, Simulated annealing procedures for forming machine cells in group technology. *European Journal of Operational Research,* **75**, 100–111.

CHENG, C. H., GOH, C. H. and LEE, A., 1996, Solving the generalized machine assignment problem in group technology. *Journal of the Operational Research Society,* **47**, 794–802.

CHU, C.-H., 1989, Cluster analysis in manufacturing cellular formation. *OMEGA: International Journal of Management Science,* **17**, 289–295; 1993, Manufacturing cell formation by competitive learning. *International Journal of Production Research,* **31**, 829–843.

COHON, J. L., 1978, *Multiobjective Programming and Planning* (New York: Academic Press).

DAHEL, N.-E. and SMITH, S. B., 1993, Designing flexibility into cellular manufacturing systems. *International Journal of Production Research,* **31**, 933–945.

GUPTA, Y. P., GUPTA, M. C., KUMAR, A. and SUNDRAM, C., 1995, Minimizing total intercell and intracell moves in cellular manufacturing: a genetic algorithm approach. *International Journal of Computer Integrated Manufacturing,* **8**, 92–101.

GUPTA, Y., GUPTA, M., KUMAR, A. and SUNDARAM, C., 1996, A genetic algorithm-based approach to cell composition and layout design problems. *International Journal of Production Research,* **34**, 447–482.

HADLEY, S. W., 1996, Finding part machine families using graph partitioning techniques. *International Journal of Production Research,* **34**, 1821–1839.

HARTLEY, R., 1983, Survey of algorithms for vector optimisation problems. In S. French *et al.* (eds) *Multi-Objective Decision Making* (London: Academic Press), pp. 1–34.

HILLIER, F. S. and LIEBERMAN, G. J., 1986, *Introduction to Operations Research* (New York: McGraw-Hill).

HO, Y.-C. and MOODIE, C. L., 1996, Solving cell formation problems in a manufacturing environment with flexible processing and routing capabilities. *International Journal of Production Research,* **34**, 2901–2923.

HWANG, C. L. and MASUD, A. S. M., 1978, *Multiple Objective Decision Making: Methods and applications* (Germany: Springer-Verlag).

HWANG, C. L. and YOON, K., 1981, *Multiple Attribute Decision Making: Methods and applications* (Germany: Springer-Verlag).

JOINES, J. A., CULBRETH, C. T. and KING, R. E., 1996, Manufacturing cell design: an integer programming model employing genetic algorithms. *IIE Transactions,* **28**, 69–85.

KAO, Y. and MOON, Y. B., 1991, A unified group technology implementation using the back-propagation learning rule of neural networks. *Computers and Industrial Engineering,* **20**, 425–437.

KAPARTHI, S. and SURESH, N. C., 1991, A neural network system for shape-based classification and coding of rotational parts. *International Journal of Production Research,* **29**, 1771–1784; 1992, Machine-component cell formation in group technology: a neural network approach. *International Journal of Production Research,* **30**, 1353–1367.

KAPARTHI, S., SURESH, N. C. and CERVENY, R. P., 1993, An improved neural network leader algorithm for part-machine grouping in group technology. *European Journal of Operational Research,* **69**, 342–356.

KEEN, P. G. W., 1977, The evolving concept of optimality. In M. K. Starr and M. Zeleny (eds) *Multiple Criteria Decision Making, TIMS studies in the Management Science,* vol. 6 (Amsterdam: North-Holland), pp. 31–57.

KING, J. R. and NAKORNCHAI, V., 1982, Machine-component group formation in group technology: review and extension. *International Journal of Production Research,* **20**, 117–133.

KLINCEWICS, J. G. and RAJAN, A., 1994, Using GRASP to solve the component grouping problem. *Naval Research Logistics,* **41**, 893–912.

KUSIAK, A., BOE, W. J. and CHENG, C. H., 1993, Designing cellular manufacturing systems: branch and bound and A* approaches. *IIE Transactions,* **25**, 46–56.

LEE, S.-D. and CHEN, Y.-L., 1997, A weighted approach for cellular manufacturing design: minimizing intercell movement and balancing workload among duplicated machines. *International Journal of Production Research,* **35**, 1125–1146.

LEE, H. and GARCIA-DIAZ, A., 1993, A network flow approach to solve clustering problems in group technology. *International Journal of Production Research,* **31**, 603–612.

LIANG, M. and TABOUN, S. M., 1995, Converting functional manufacturing systems into focused machine cells–a bicriterion approach. *International Journal of Production Research,* **33**, 2147–2161.

LOGENDRAN, R., 1993, A binary integer programming approach for simultaneous machine-part grouping in cellular manufacturing systems. *Computers and Industrial Engineering,* **24**, 329–336.

MIN, H. and SHIN, D., 1993, Simultaneous formation of machine and human cells in group technology: a multiple objective approach. *International Journal of Production Research,* **31**, 2307–2318.

MOSIER, C. and TAUBE, L., 1985, The facets of group technology and their impacts on implementation–a state-of-the-art survey. *OMEGA: International Journal of Management Science,* **13**, 381–391.

MUKHOPADHYAY, S. K., SARKAR, P. and PANDA, R. P., 1994, Machine-component grouping in cellular manufacturing by multidimensiontal scaling. *International Journal of Production Research,* **32**, 457–477.

OFFODILE, O. F., MEHREZ, A. and GRZNAR, J., 1994, Cellular manufacturing: a taxonomic review framework. *Journal of Manufacturing Systems,* **13**, 196–220.

RAJAMANI, D., SINGH, N. and ANEJA, Y. P., 1996, Design of cellular manufacturing systems. *International Journal of Production Research,* **34**, 1917–1928.

SANKARAN, S., 1990, Multiple objective decision making approach to cell formation: a goal programming model. *Mathematical and Computer Modeling,* **13**, 71–81.

SANKARAN, S. and KASILINGAM, R. G., 1993, On cell size and machine requirements planning in group technology. *European Journal of Operational Research,* **69**, 373–383.

SHAFER, S .M., KERN, G. M. and WEI, J. C., 1992, A mathematical programming approach for dealing with exceptional elements in cellular manufacturing. *International Journal of Production Research,* **30**, 1029–1036.

SHAFER, S. M. and ROGERS, D. F., 1991, A goal programming approach to the cell formation problem. *Journal of Operations Management,* **10**, 28–43.

SOFIANOPOULOU, S., 1997, Application of simulated annealing to a linear model for the formulation of machine cells in group technology. *International Journal of Production Research,* **35**, 501–511.

SONG, S. and HITOMI, K., 1992, GT cell formation for minimising the intercell parts flow. *International Journal of Production Research,* **30**, 2737–2753.

SU, C.-T. and HSU, C.-M., 1998, Multi-objective machine-part cell formation through parallel simulated annealing. *International Journal of Production Research,* **36**, 2185–2207.

SURESH, N. C. and KAPARTHI, S., 1994, Performance of fuzzy ART neural network for group technology cell formation. *International Journal of Production Research,* **32**, 1693–1713.

SURESH, N. C., SLOMP, J. and KAPARTHI, S., 1995, The capacitated cell formation problem: a new hierarchical methodology. *International Journal of Production Research,* **33**, 1761–1784.

SZIDAROVSKY, F., GERSHON, M. E. and DUKSTEIN, L., 1986, *Techniques for Multiobjective Decision Making in Systems Management* (New York, NY: Elsevier).

VANELLI, A. and HALL, R. G., 1993, An eigenvector solution methodology for finding part-machine families. *International Journal of Production Research,* **31**, 325–349.

VENUGOPAL, V. and NARENDRAN, T. T., 1992a, A genetic algorithm approach to the machine-component grouping problem with multiple objectives. *Computers and Industrial Engineering,* **22**, 469–480; 1992b, Cell formation in manufacturing systems through simulated annealing: An experimental evaluation. *European Journal of Operational Research,* **63**, 409–422.

WEI, J. C. and GAITHER, N., 1990, A capacity constrained multiobjective cell formation method. *Journal of Manufacturing Systems,* **9**, 222–232.

WEMMERLÖV, U. and HYER, N. L., 1986, Procedures for the part family/machine group identification problem in cellular manufacturing. *Journal of Operations Management,* **6**, 125–147.

ZELENY, M., 1982, *Multiple Criteria Decision Making* (New York, NY: McGraw-Hill).