Objective Function Effect Based Pattern Search—An Implementation for 3D Component Layout

Chandankumar Aladahalli
e-mail: chandan@alumni.cmu.edu

Jonathan Cagan[1]
e-mail: cagan@cmu.edu

Kenji Shimada
e-mail: shimada@cmu.edu

Department of Mechanical Engineering,
Carnegie Mellon University,
5000 Forbes Avenue,
Pittsburgh, PA

Generalized pattern search (GPS) algorithms have been used successfully to solve three-dimensional (3D) component layout problems. These algorithms use a set of patterns and successively decreasing step sizes of these patterns to explore the search space before converging to good local minima. A shortcoming of conventional GPS algorithms is the lack of recognition of the fact that patterns affect the objective function by different amounts and hence it might be efficient to introduce them into the search in a certain order rather than introduce all of them at the beginning of the search. To address this shortcoming, it has been shown by the authors in previous work that it is more efficient to schedule patterns in decreasing order of their effect on the objective function. The effect of the patterns on the objective function was estimated by the a priori expectation of the objective function change due to the patterns. However, computing the a priori expectation is expensive, and to practically implement the scheduling of patterns, an inexpensive estimate of the effect on the objective function is necessary. This paper introduces a metric for geometric layout called the sensitivity metric that is computationally inexpensive, to estimate the effect of pattern moves on the objective function. A new pattern search algorithm that uses the sensitivity metric to schedule patterns is shown to perform as well as the pattern search algorithm that used the a priori expectation of the objective function change. Though the sensitivity metric applies to the class of geometric layout or placement problems, the foundation and approach is useful for developing metrics for other optimization problems. [DOI: 10.1115/1.2406096]

Keywords: pattern search, 3D component layout, expected change in objective function, sensitivity metric

1 Introduction

Three-dimensional (3D) component layout is a framework for formulating and solving the problem of determining the optimal spatial location and orientation of a set of components given an objective function and constraints in mechanical and electromechanical products [1]. Applications of the 3D component layout problem include layout of automobile engine compartments [2], automobile trunk packing for maximizing luggage space [3], transmission layout [4], and layout and routing of a heat pump [2], and almost any mechanical or electro-mechanical machine whose components need to be placed to satisfy constraints and optimize one or more objectives.

Generalized pattern search (GPS) algorithms have been successfully applied to 3D component layout problems with modifications and extensions [2]. GPS algorithms are greedy direct search methods that use moves along a set of search directions called patterns and successively decreasing step sizes of these patterns to explore the search space and converge to good local minima [5]. In 3D component layout, the patterns are typically the translations and rotations of objects and the step sizes are the amount of translation, and rotation, respectively.

A drawback of the GPS algorithm is that it applies all patterns from the beginning of the search and decreases their step size at the same rate without recognizing the fact that different patterns affect the objective function by different amounts and hence it might be useful to apply them according to some schedule. To remedy this drawback, an efficient schedule that introduces patterns in decreasing order of their effect on the objective function was presented in Ref. [6]. The motivation behind this schedule is to prevent moves with smaller effects on the objective function from being applied with moves with larger effects on the objective function. Doing so prevents the moves with smaller effect from being wasted since the small improvements in the objective function due to them are overridden by large improvements in the objective function due to those with larger effects when applied together. The pattern search algorithm that uses this schedule is called the objective function-based pattern search (OPS).

The OPS algorithm is able to improve run time by 30% on average over the GPS algorithm [6]. The effect on the objective is quantified as the expected change in objective function value due to the moves of patterns. Computing the actual expected change in objective function is very expensive and offsets the subsequent savings in runtime. The OPS algorithm served the purpose of demonstrating that applying moves in decreasing order of their effect on the objective function is indeed useful. To make the OPS algorithm practical to implement, we introduce the sensitivity metric in this paper as a substitute for the expected change in objective function value to estimate the effect on the objective function.

The sensitivity metric is a combination of two quantities when a move is applied. The first quantity is the amount of nonintersection between the component before the move and itself after the move. The second quantity is the amount of displacement of the component. The new algorithm using the sensitivity metric is called the metric objective-function-based pattern search (MOPS)

[1]Corresponding author.

Contributed by the Design for Manufacturing Committee of ASME for publication in the Journal of Mechanical Design. Manuscript received April 11, 2005; final manuscript received December 21, 2005. Review conducted by David Kazmer.

algorithm. It is implemented by replacing the OPS algorithm's mapping between the step sizes and the expected change in objective function value with a mapping between the step sizes and the sensitivity metric. Computing the sensitivity metric mapping is about two orders of magnitude less expensive than computing the expected change in objective function mapping, as shown in Sec. 5.2, and is negligible compared to the runtime of the subsequent pattern search algorithm. The MOPS algorithm performs as well as the original OPS algorithm and yields similar improvements in run time over the GPS algorithm when applied to 3D component layout problems.

The remainder of the paper is organized as follows. Section 2 presents the background on the 3D component layout problem, and summarizes the OPS algorithm that schedules patterns in decreasing order of their effect on the objective function. Section 3 presents the motivation for developing the new sensitivity metric. Section 4 presents the sensitivity metric. Section 5 presents the sensitivity metric for three different 3D component layout problems and compares them to the expected change in objective function value. Section 6 presents the results of solving 3D component layout problems using the MOPS algorithm and compares them with the results obtained in Ref. [6] using the GPS and OPS algorithms. Results comparing the MOPS and the GPS algorithms on two new examples: trunk packing and printed circuit board layout problem are also presented. Section 7 presents discussion and conclusions.

2 Background

3D component layout problems have been successfully solved by pattern search algorithms [1–4]. This section briefly discusses the 3D component layout problem, the pattern search methods used to solve it, and the OPS algorithm.

2.1 3D Component Layout. Many mechanical and electromechanical products are a combination of functionally and geometrically inter-related components. The spatial location and orientation of these components affect a number of physical quantities of interest to the designer, engineer, manufacturer and the end user of the product. The 3D component layout framework concerns itself with determining the optimal spatial location and orientation of a set of components given some design objectives and constraints. It models the layout problem as a minimization problem with the objective function being a weighted sum of the design objectives and penalties for constraint violation. The design objectives can include a quantification of a variety of measures such as the amount of cable used in the engine compartment of a car, or the component packing density in an electric drill, or the center of gravity of a space vehicle. The most significant constraint is the non-intersection of components and non-protrusion of components outside the design space. Other constraints include spatial relationships between components and between a component and the design space.

Many different stochastic search algorithms have been applied to the 3D component layout optimization problem [7]. These include genetic algorithms [8–11], simulated annealing [12–16], and generalized pattern search based algorithms [2–4]. The current algorithm of choice is the extended pattern search algorithm (EPS) [2], which is an extension of the GPS. GPS algorithms use moves that are steps of varying sizes of a set of search directions called patterns, to explore the search space. In the 3D component layout framework, the patterns are translations and rotations of components being placed and step sizes are the amount of translations and rotations. The GPS algorithm starts by applying moves of large step sizes of all patterns. Only moves resulting in an improved objective function are accepted. Moves of the patterns are reapplied until none of them is able to improve the objective function. At this point step sizes of all patterns are scaled down by a same factor and the search proceeds with the new step sizes of patterns. The search terminates when the step sizes become smaller than a user-defined threshold. To address some deficiencies regarding the pattern selection and step size decrease in the GPS algorithm, the objective function-based pattern search algorithm was developed by Aladahalli et al. [6].

2.2 Objective Function-based Pattern Search. The OPS algorithm [6] was motivated by the observation that although different patterns affect the objective function by different amounts, the GPS algorithm applies all the patterns right from the beginning of the search. In the early stages of search the moves of patterns resulting in smaller improvements in objective function are overridden by those resulting in larger improvements and are hence wasted. Also, in the GPS algorithm, smaller improvements earlier in the search force the algorithm to reapply moves of all the patterns again resulting in an increased number of iterations to converge. This is because GPS algorithms only decrease step size when none of the patterns result in an improvement in the objective function at the current step size.

The OPS algorithm addresses the above deficiency in the GPS algorithm by scheduling patterns in decreasing order of their effect on the objective function value. While moves of patterns with big effects on the objective function start being applied right at the beginning of the search, the moves of patterns with smaller effects are introduced later on. Towards the end of the search all patterns are active. This contrasts with the GPS algorithm which starts applying moves of all patterns right from the beginning and all patterns are active at all times during the search. Also, the OPS algorithm decreases the step sizes of patterns in inverse proportion to the rate of decrease of their effect on the objective function. This is done so that the effect on the objective function decreases by the same amount for the different active patterns. This ensures that pattern step sizes are such that they have the same effect on the objective function at all stages of the search and hence mixing of moves with smaller and bigger effects on the objective function is avoided. The GPS algorithm in contrast decreases all pattern step sizes at the same rate, wasting moves towards the start. Figure 1 contrasts the OPS and GPS algorithms.

While the GPS algorithm starts with the moves corresponding to the biggest step size of all patterns and proceeds with the search by successively applying moves with decreasing step size of patterns, the OPS algorithm starts with moves corresponding to the largest effect on the objective function, and proceeds with the search by successively applying moves with decreasing effect on the objective function. Thus, the OPS algorithm drives pattern search by the effect of patterns on the objective function.

In its implementation, the OPS algorithm first quantifies the effect on the objective function due to a move as the expected change in the objective function value, E, due to the application of that move. It uses a mapping from step sizes to E for every pattern to schedule them in the search. To calculate E, the move is applied from a large number (50 in Ref. [6]) of random initial placements of components and the average of the changes in objective function is computed to give the expected change in the objective function value. Thus computing the mapping is very expensive since it involves multiple computations of the objective function. As pointed out in Ref. [6], this preprocessing step can be as expensive as the time savings over the GPS algorithm. Inexpensively estimating the effects of patterns on the objective function was the motivation to look for alternate metrics as discussed in the following section.

3 Motivation

The OPS algorithm demonstrated that scheduling patterns in decreasing order of their effect on the objective function value is indeed effective. It uses the expected change in objective function value to estimate the effect of the patterns on the objective function. However, for a practical implementation an inexpensive estimate of the effect on the objective function is required. For constructing an inexpensive metric, it is important to note that the

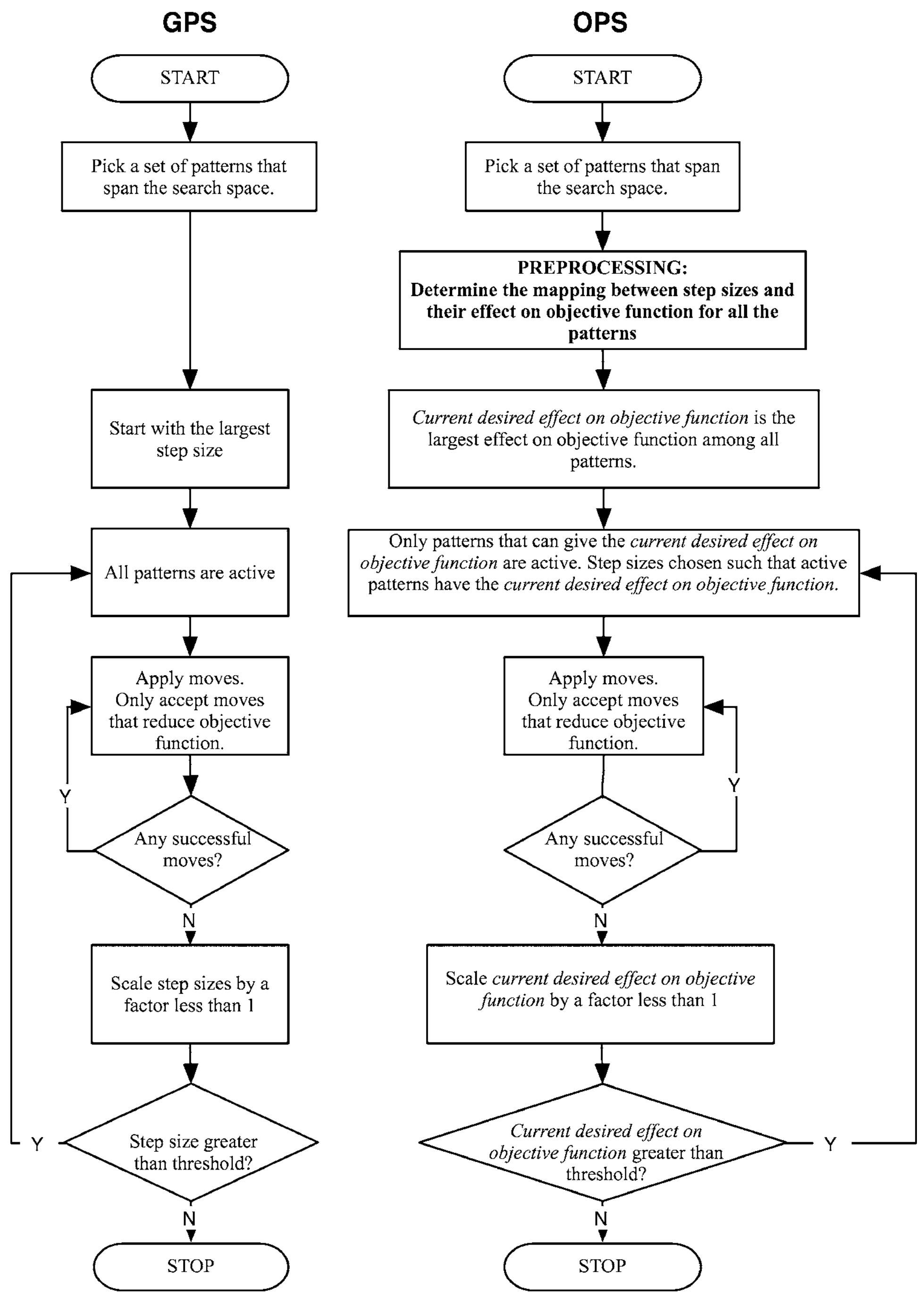

GPS
OPS
START
Pick a set of patterns that span the search space.
PREPROCESSING: Determine the mapping between step sizes and their effect on objective function for all the patterns
Start with the largest step size
Current desired effect on objective function is the largest effect on objective function among all patterns.
All patterns are active
Only patterns that can give the current desired effect on objective function are active. Step sizes chosen such that active patterns have the current desired effect on objective function.
Apply moves. Only accept moves that reduce objective function.
Any successful moves?
Y
N
Scale step sizes by a factor less than 1
Scale current desired effect on objective function by a factor less than 1
Step size greater than threshold?
Current desired effect on objective function greater than threshold?
STOP

Fig. 1 A comparison between the GPS and OPS algorithms. The OPS algorithm drives pattern search by the effect on the objective function due to different patterns.

scheduling of patterns in the OPS algorithm uses the relative effects on the objective function of the patterns and not the absolute effects.

Figure 2 illustrates the mapping between step size and expected change in objective function value for two patterns. It can be seen that Pattern 1 has a higher effect ($E1_{\max}$) on the objective function at its maximum step size than Pattern 2 ($E2_{\max}$). Therefore, in the OPS algorithm moves of Pattern 1 start before moves of Pattern 2. The absolute values $E1_{\max}$ and $E2_{\max}$ are not directly used by the OPS algorithm. Rather, it uses the relative values of $E1_{\max}$ and $E2_{\max}$ to decide what pattern is introduced first. An inexpensive metric should capture the relative maximum effects of the patterns on the objective function.

Also, in the OPS algorithm, the relative rates at which the pattern step sizes are decreased depends inversely on the relative rates of decrease of the effect on the objective function with respect to the step sizes of those patterns. For example in Fig. 2 the effect on the objective function decreases more per unit step size for Pattern 1 than Pattern 2. Therefore, the OPS algorithm decreases the step size of Pattern 2 faster than that of Pattern 1. Again, an inexpensive metric should reflect the relative rates of decrease in the effect on the objective function for different patterns.

To summarize, we seek an inexpensive metric that captures the relative information about patterns' effects on the objective function. We propose that such metrics can be constructed in most

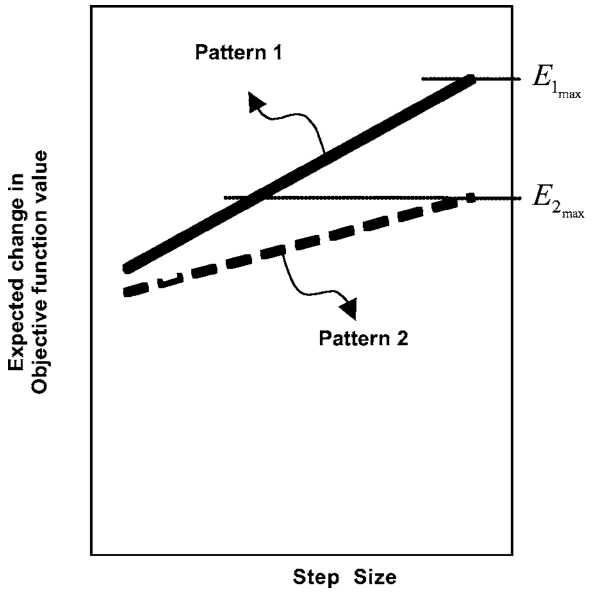
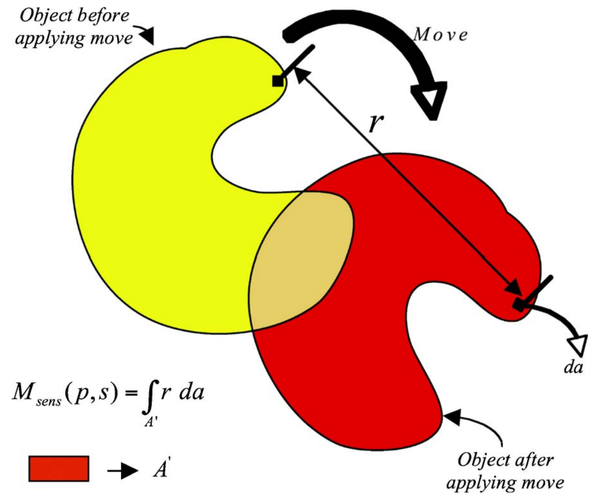


Fig. 2 The relative nature of the information used by OPS



Object before applying move
Move
r
da
$M_{sens}(p,s) = \int_{A'} r\, da$
🟥 → A'
Object after applying move

Fig. 3 The sensitivity metric

domains using domain knowledge and in this paper we develop one for component layout. In 3D component layout, geometry and size of components can be used to estimate the relative effects of moves on the objective function. For example, a move associated with a bigger component will have a bigger effect on the intersection and protrusion volume, than a smaller component. Also, larger translations of a component have a bigger effect on the intersection and protrusion volume in particular and layout objectives in general. From observations such as these, the sensitivity metric was developed to estimate the effect of pattern moves on the objective function and use the estimations to schedule patterns in the MOPS algorithm.

4 The Sensitivity Metric

Most design objectives in 3D component layout depend on the spatial location, geometry and size of the components being placed. In particular, one objective, namely the amount of pairwise intersections between components, is directly related to the location, geometry and size of the components. This objective results from enforcing the nonintersection constraint between components by penalizing intersections. This objective is a significant part of the 3D component layout objective function because, as a penalty term, it is weighed heavily to push the intersections between components to zero. The sensitivity metric proposed in this section estimates the effect on such objectives due to a move by using the geometry of the component that is moved and the nature and magnitude of move. In the following subsection, required characteristics of the sensitivity metric are derived based on observations of the effect of pattern moves on 3D component layout objectives. The characteristics are then used to build a definition of sensitivity that embodies them.

4.1 Required Characteristics of the Sensitivity Metric. Recognizing that the 3D component layout objective functions depend on the location, geometry and size of the components, the following observations can be made. The first observation relates to the geometry and size of the component and the direction of the move (pattern). For similar displacements, the effect on the objective function is proportional to the nonintersection volume between the component before the move and itself after the move. For example, consider moves along two patterns of a cylinder of diameter one unit and length ten units. The first move is a translation of two units along the radial direction, and the second move is a translation of two units along the axial direction. Both moves have the same step size, and the volume of the components being displaced is the same. But the first displacement has a potential for bigger effect on the objective than the second. This is because the first displacement causes the cylinder to be displaced to a location completely not occupied by itself before the move, whereas the second displacement moves only 20% of the volume of the cylinder to a location not occupied by itself before the move.

The second observation relates to the magnitude of the moves (step sizes of patterns). The effect of a move on the objective function depends on the amount of displacement of components. But, in the case of rotations, a larger rotation does not guarantee a bigger change in objective function. For example, rotating a spherical component about an axis through its centroid does not affect an objective function such as the penalty for component–component intersections, even though the individual volume elements in the sphere are displaced. A move affects the objective function only to the extent that it results in actual displacement of the component to a location not occupied by it previously.

The third observation relates to the translation and rotation patterns. Since pattern search for 3D component layout relies on both translation and rotation patterns, the metric must be able to measure both their effects on the objective function.

From the above observations, a metric that will substitute the expected change in objective function value must have the following characteristics: (1) it must be proportional to the amount of nonintersection created between the component before applying the move and itself after applying the move; (2) it must be proportional to the amount of displacement of a component to a location not occupied by it before the move; and (3) it must be applicable to translations, rotations and combinations of translations and rotations.

4.2 Sensitivity Metric Definition. We propose the following definition for a sensitivity metric so that it embodies the characteristics required in Sec. 4.1. The sensitivity metric, M_{sens}, for a pattern p at a step size s is defined as follows

$$M_{\text{sens}}(p,s) = \int_{V'} r\, dv \tag{1}$$

where r is the displacement of the infinitesimal volume dv and V' is the nonintersecting volume between a component before applying the move and itself after applying the move. In 2D the sensitivity metric is analogously defined as follows

$$M_{\mathrm{sens}}(p,s) = \int_{A'} r\, da \qquad (2)$$

where r is the displacement of the infinitesimal area da, and A' is the nonintersecting area between a component before applying the move and itself after applying the move. Figure 3 illustrates a 2D example.

In 3D component layout, the integral in Eq. (1) is evaluated by taking advantage of the already available octree representation of components [17]. In our implementation it is evaluated as a discrete sum over the voxels of a component using its octree decomposition. For rotations where the amount of nonintersecting volume between a component before applying the move and itself after applying the move may be periodic with respect to the step size, we compute and use the sensitivity only over the first instance of the period.

Since the integral in Eq. (1) is evaluated only over the nonintersecting volume created between the object before applying the move and itself after applying the move it has the first characteristic required in Sec. 4.1. Since the integral is evaluated only over the non-intersecting volume, and the integrand is a product of the displacement of the volume element and its volume, displacements of volume elements that do not result in non-intersection are excluded from the sensitivity metric. Thus the sensitivity metric has the second required characteristic from Sec. 4.1.

By computing the sensitivity metric as an integral over volume elements, rotation moves can be decomposed into a collection of translations of infinitesimal volume elements. Thus, the metric provides a common definition to capture the effects of translations and rotations and combinations of translations and rotations on the objective function—the third requirement on the sensitivity metric from Sec. 4.1. In fact, the definition is applicable to any shape preserving transformation since it reduces a transformation into a collection of translations of infinitesimal volume elements.

We emphasize here that the above definition is not unique, but it serves the purpose of inexpensively estimating relative effects of different moves on 3D component layout objective functions. Though the above definition is specific to general 3D component layout objective functions, we believe metrics containing relative information similar the expected change in objective function value can be derived from domain knowledge in other optimization problems as well.

5 Comparison Between the Sensitivity Metric and Expected Change in Objective Function Value

The sensitivity metric was computed for two examples each for the three 3D component layout problems described in Ref. [6]. The three problems are: packing into spherical container, packing into stereolithography apparatus (SLA) while minimizing maximum height of the packing as defined in Ref. [18] and packing a set of objects while minimizing the moment of the packing about its centroid. The respective objective functions were: (1) sum of pairwise intersections between components and protrusions of components outside the container; (2) sum of intersections and protrusions and the maximum height of the packing; and (3) sum of intersections and protrusions and the second moment of the packing about the centroid of the packing. The first example had 13 components and the second had 18 components in the packing.

5.1 Comparison Between the Relative Information in the Sensitivity Metric and the Expected Change in Objective Function. As mentioned in Sec. 3, the OPS algorithm uses the relative effect on objective function between patterns to schedule them. This section compares the relative effect on the objective function due to patterns by using the sensitivity metric and the relative effect on the objective function due to the patterns by using the expected change in objective function. Figure 4 compares the mapping between step size and the sensitivity metric and the mapping between step size and the expected change in objective function value for a few patterns in the first example for each of the three 3D component layout problems mentioned above. The same set of components was packed in the three problems and hence they have the same patterns. All four plots are on a log–log scale. The different dotted lines indicate the sensitivities plotted against step size *(a)* or the expected change in objective function plotted against the step size *(b)*,*(c)*, and *(d)*. Since the sensitivity metric is independent of the objective function, its value is the same for the three different 3D component layout problems. The plots were similar for the second example which had 18 objects in the packing.

It must be noted that the plots shown in Fig. 4 are not directly used in the OPS or the MOPS algorithms. Instead, they are approximated by a linear interpolation (on the log–log scale) between the maximum and minimum values, since a linear approximation is a good fit and provides an elegant analytical mapping. Also, it helps in reducing the computation since with the linear interpolation only the maximum and minimum values need to be calculated.

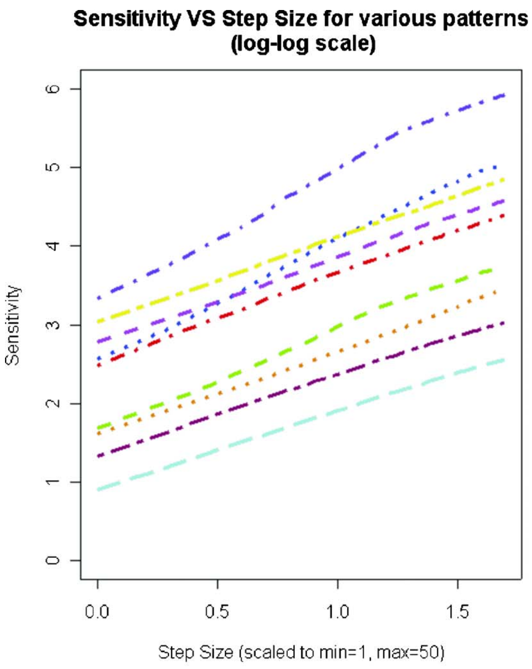
From Fig. 4 it can be seen that the sensitivity metric captures the relative effects on the objective function due to different patterns quite similarly to the expected change in objective function. The order of the patterns by their highest sensitivity values (corresponding to the largest step sizes) is similar to the order of the patterns by their highest expected change in objective function values. Also the relative slopes of the sensitivity metric plots for the different patterns are similar to the relative slopes of the expected change in objective function value plots for the patterns.

For a more thorough comparison between the relative information contained in the sensitivity metric mapping and the expected change in objective function mapping, we computed how well the two mappings matched according to two criteria based on qualitative similarity to each other for the two examples in the three layout problems.

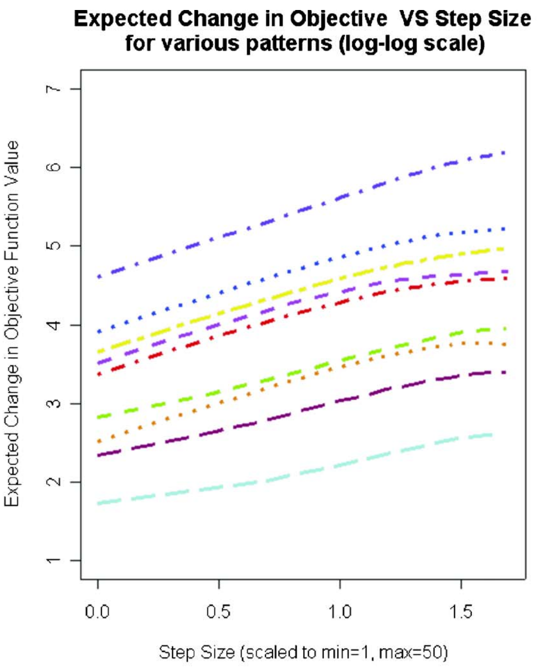
Given two patterns, p_i and p_j, the first criterion was whether the relative starting values of both the sensitivity and the expected change in objective function value had a similar qualitative relation to each other. For example, if the starting value of the expected change in objective function value of p_i was greater than the starting value of the expected change in objective function value of p_j, then it was checked whether the starting value of the sensitivity of p_i was also greater than the starting value of sensitivity of p_i. This was checked for all possible combination of patterns. This criterion reflects whether the order in which patterns are introduced is the same according to both the mappings. In this example, moves of pattern p_i are introduced before moves of pattern p_j. The results for two packing examples each of the three layout problems are shown in Table 1. More than 90% of pattern pairs have the same order of starting sensitivity as the order of expected change in objective function.

The second criterion was whether pairs of patterns have the same qualitative relation in their slopes in both the sensitivity and expected change in objective function mappings. For example if the slope of the expected change in objective function value of p_i was greater than the slope of the expected change in objective function value of p_j then it was checked whether the slope of the sensitivity of p_i was also greater than the slope of sensitivity of p_j. This criterion checks if the relative rates at which the pattern step sizes are decreased are the same qualitatively according to both the mappings. In this example, the step size pattern p_j will be decreased faster than that of p_i. The results are shown in Table 1. Here the percentage of pattern pairs satisfying the criteria is about 75%.

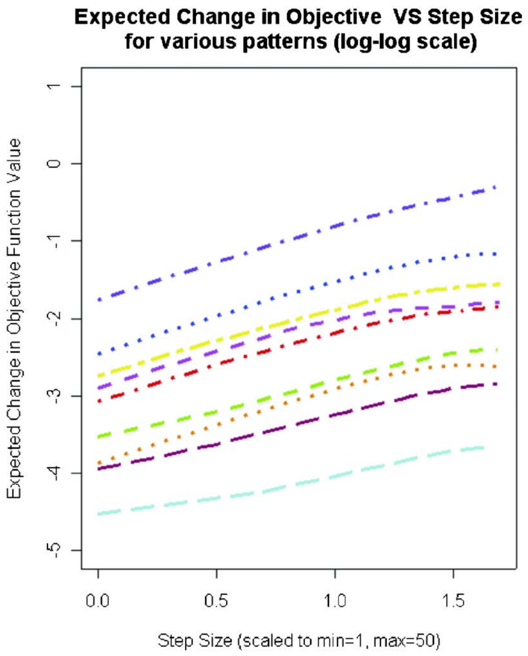
By the above two criteria, and mappings such as those shown in Fig. 4, we conclude there is a good similarity in the qualitative information contained in the sensitivity mapping and the expected change in objective function mapping.



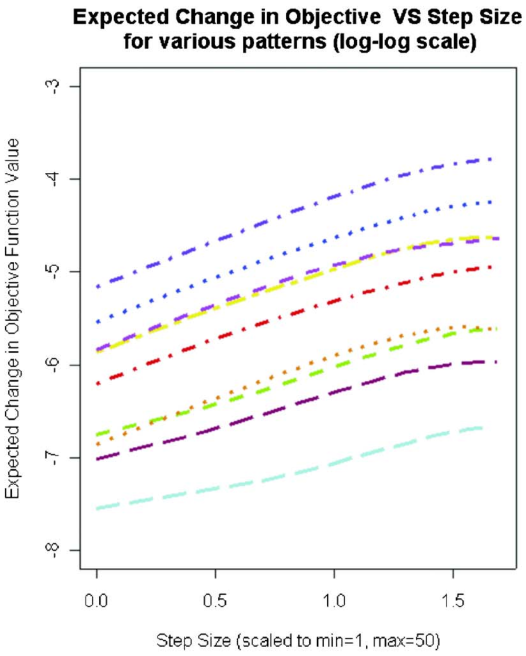
(a) Sensitivities for the three 3D component layout problems



(b) Expected change in objective function value for packing into a spherical container



(c) Expected change in objective function value for packing into a SLA container minimizing height



(d) Expected change in objective function value for packing minimizing moment about centroid

Fig. 4 Comparison of the sensitivity metric: (*a*) with expected change in objective function value for three different layout problems (*b*), (*c*), and (*d*)

5.2 Computational Cost Comparison for the Sensitivity Metric and the Expected Change in Objective Function. For preprocessing, a MOPS algorithm needs to compute the sensitivity metric at the minimum and maximum step size of every pattern. Therefore for a 3D component layout problem with n patterns, it needs $2n$ evaluations of the sensitivity metric. Similarly for preprocessing, the OPS algorithm needs to compute the expected change in objective function value at the minimum and maximum step size of every pattern. Therefore $2n$ evaluations of the expected change in objective function value are required. To compute the expected change in objective function value the average of 50 values of the change in objective function due to the move application is computed. Each change in objective function needs two evaluations of the objective function. Thus a total of 200 evaluations of the objective function is required for the determining the expected change in objective function for a pattern.

Table 1 Percentage of pattern pairs satisfying the two matching criteria

	Example	Criterion 1 (%)	Criterion 2 (%)
Packing in sphere	1	94	80
	2	89	77
Packing in SLA container	1	93	75
	2	93	74
Packing to minimize *I*	1	93	75
	2	92	73

Therefore for a 3D component layout problem with *n* patterns, the preprocessing step in the OPS algorithm requires 200*n* objective function evaluations, a factor of 100 greater than MOPS.

Also, computing the sensitivity metric is less expensive than computing the objective function. This is because it involves computing only one intersection, whereas computing the objective function involves computing the intersection of the component with all the other components and the container. Computing the objective function may also involve evaluation of additional terms in the objective function (like maximum height, moment of inertia, etc.). Thus computing the sensitivity mappings is more than two orders of magnitude less expensive than computing the expected change in objective function value mappings.

6 Experiments and Results

The MOPS algorithm was implemented by replacing the mapping between the step sizes and the expected change in objective function value with a mapping between the step sizes and the sensitivity metric in the OPS algorithm.

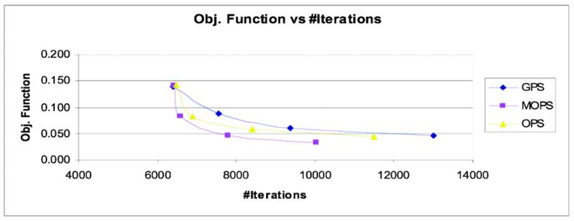
The two examples of the three 3D component layout problems mentioned in Sec. 5 were solved with the new MOPS algorithm and their results were compared to those obtained using the GPS and OPS algorithms. Also, two other layout problems were solved with the MOPS and the GPS algorithms to show the effectiveness of the new MOPS algorithm in reducing run time over the GPS algorithm. The first layout problem involved packing SAE standard luggage pieces into automobile trunks [3]. The objective minimized was the sum of pairwise intersections and protrusions out of the trunk. The second layout problem involved the placement of components on a printed circuit board (PCB). The objective was to minimize the sum of pairwise intersections and protrusions and a measure of wiring length between the circuit elements.

6.1 Comparison Between MOPS, OPS, and GPS Algorithms on 3D Component Layout Problems.

The results from the first example of the three layout problems using the MOPS algorithm are shown in Fig. 5. The results are compared with the results obtained by the GPS and OPS algorithms from Ref. [6]. The four rows in the tables in the figure are the results for four different lengths of runs of the algorithms. The objective function values are the average of 40 best of three runs of the algorithm. A best of three runs solution is the best objective function value obtained by running the algorithm three times from different random starting points. The number of iterations shown is the average value for a single run. It can be seen that the MOPS algorithm performs 30% faster than the GPS algorithm on average. The results were similar for the second example for the three layout problems. This improvement in performance is similar to the one obtained by the OPS algorithm in Ref. [6]. But the MOPS algorithm does not require the expensive preprocessing step of evaluating the mapping between the expected change in objective function values and the step sizes for the patterns.

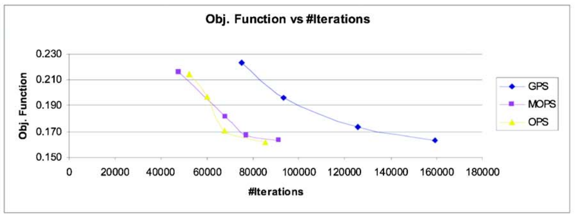
The number of sensitivity evaluations for preprocessing in MOPS is compared with the number of objective function evaluations for preprocessing in OPS in Table 2. As shown in Sec. 5.2, the number of iterations required by the MOPS algorithm is two orders of magnitude smaller than that for the OPS algorithm. Comparing the number of sensitivity evaluations in Table 2 to the number of objective function evaluations (No. Iter) in Fig. 5 it can be seen that on average the time required for preprocessing by the MOPS is less than 1% of the total iterations needed by the search. It is to be noted that typically a best of three runs solution is used. Therefore the total number of objective function evaluations for the search is three times the number shown Fig. 5. The negligible

GPS		MOPS		OPS		Imp (%) MOPS		Imp (%) OPS	
Obj.Fn.	#Iter	Obj.Fn.	#Iter	Obj.Fn.	#Iter	Obj.Fn.	#Iter	Obj.Fn.	#Iter
0.140	6,401	0.141	6,431	0.144	6,466	-1.2	-0.5	-3.1	-1.0
0.089	7,562	0.082	6,600	0.082	6,901	7.4	12.7	7.8	8.7
0.061	9,364	0.046	7,803	0.058	8,421	24.4	16.7	3.7	10.1
0.047	13,006	0.032	10,032	0.043	11,487	30.5	22.9	6.8	11.7



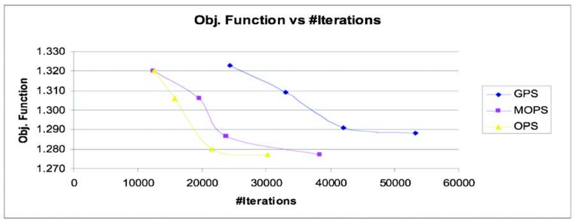
Packing into sphere – Example 1

GPS		MOPS		OPS		Imp (%) MOPS		Imp (%) OPS	
Obj.Fn.	#Iter	Obj.Fn.	#Iter	Obj.Fn.	#Iter	Obj.Fn.	#Iter	Obj.Fn.	#Iter
0.223	75,183	0.216	47,567	0.215	52,433	3.3	36.7	3.9	30.3
0.196	93,518	0.182	68,193	0.197	60,189	7.3	27.1	-0.5	35.6
0.173	125,836	0.167	77,275	0.170	67,759	3.7	38.6	1.7	46.2
0.163	159,415	0.163	91,394	0.162	85,293	0.0	42.7	0.7	46.5



Packing into SLA container – Example 1

GPS		MOPS		OPS		Imp (%) MOPS		Imp (%) OPS	
Obj.Fn.	#Iter	Obj.Fn.	#Iter	Obj.Fn.	#Iter	Obj.Fn.	#Iter	Obj.Fn.	#Iter
1.323	24,319	1.320	12,391	1.320	12,553	0.2	49.0	0.2	48.4
1.309	33,078	1.306	19,600	1.306	15,768	0.2	40.7	0.2	52.3
1.291	42,012	1.287	23,712	1.280	21,543	0.3	43.6	0.9	48.7
1.288	53,231	1.277	38,300	1.277	30,188	0.9	28.0	0.9	43.3



Packing to minimize moment of inertia about centroid – Example 1

Fig. 5 Results comparing the GPS and MOPS algorithms for the three packing problems

Table 2 Preprocessing times for the MOPS and OPS algorithms

	Example	No. sensitivity evaluations for MOPS preprocessing	No. objective function evaluations for OPS preprocessing
Packing in sphere	1	156	15,600
	2	216	21,600
SLA packing	1	156	15,600
	2	216	21,600
Minimizing moment of intertia	1	156	15,600
	2	216	21,600

pre-processing time makes the complete MOPS algorithm, including the preprocessing step, run faster than the GPS algorithm with better results.

The definition of sensitivity metric presented above is not unique. It is possible to use other metrics that estimate the effect of a pattern move on the objective function. The sensitivity metric was compared with two other similar metrics M'_{sens} and M''_{sens}

$$M'_{\text{sens}}(p,s) = \int_{V'} \sqrt{r}\, dv$$

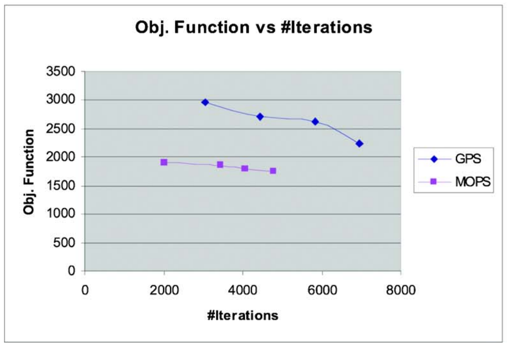
$$M''_{\text{sens}}(p,s) = \int_{V'} r^2\, dv$$

These two metrics are modifications of the sensitivity metric and were constructed to vary the relative importance of intersection avoidance and displacement distance due to a move. One increases the importance of the distance term in the sensitivity metric by using a r^2 instead of a r term. The other decreases the importance of the distance term in the sensitivity term by using a $\sqrt{r}$ term and thus increases the importance of the intersection volume avoidance. On the six 3D component layout examples discussed above, the two metrics M'_{sens} and M''_{sens} performed similarly or up to 10% worse than the sensitivity metric. We believe that by using a linear r term, the sensitivity metric strikes a good balance between accounting for intersection avoidance of a move and the amount by which it displaces a component both of which contribute to affecting the objective function value.

6.2 Comparison on Trunk Packing Problems. To further explore the effectiveness of scheduling patterns in the MOPS algorithm over the GPS algorithm, the two algorithms were compared on two automobile trunk packing instances [3]. Here the objective was to pack a preselected set of luggage pieces from the SAE standard luggage set into an automobile trunk. This is a modification of the problem in Ref. [3] where the subset of lug-

EXAMPLE 1

GPS		MOPS		Imp (%)	
Obj.Fn.	#Iter	Obj.Fn.	#Iter	Obj.Fn.	#Iter
2965	3,027	1894	2,023	36.1	33.1
2702	4,438	1861	3,449	31.1	22.2
2631	5,814	1800	4,050	31.5	30.3
2247	6,934	1759	4,787	21.7	30.9



EXAMPLE 2

GPS		MOPS		Imp (%)	
Obj.Fn.	#Iter	Obj.Fn.	#Iter	Obj.Fn.	#Iter
2642	2,959	2629	1,800	0.4	39.1
2544	4,482	2423	2,010	4.7	55.1
2454	5,744	2361	3,252	3.7	43.3
1833	6,767	1849	4,033	-0.8	40.4

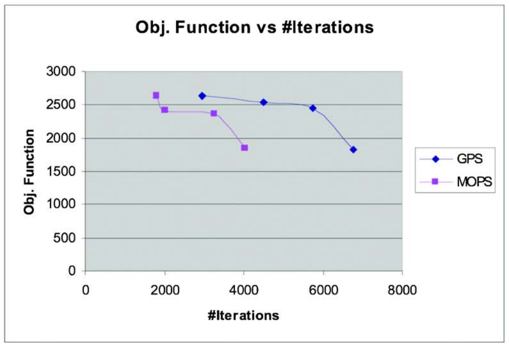
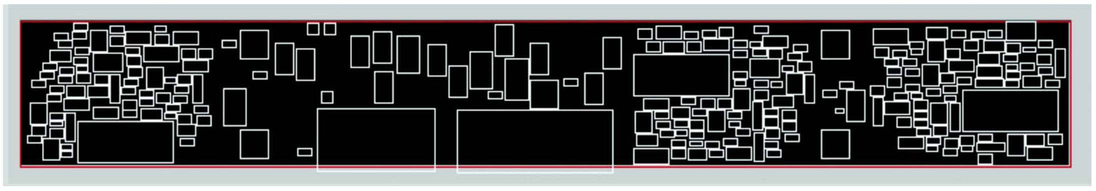


Fig. 6 Results comparing the GPS and MOPS algorithms on two trunk packing examples



GPS		MOPS		Imp (%)	
Obj.Fn.	#Iter	Obj.Fn.	#Iter	Obj.Fn.	#Iter
11.42	52,609	5.21	48,279	54.3	8.2
10.52	68,077	5.04	60,690	52	10.8
10.02	124,399	4.92	86,106	50.9	30.7
9.72	177,056	4.83	114,876	50.3	35.1

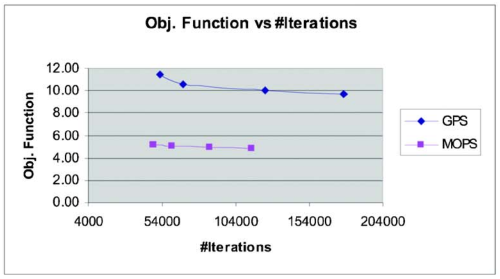


Fig. 7 Results comparing the GPS and the OPS algorithms for the layout of a PCB board

gage pieces to be packed had to be chosen by the packing algorithm. Two instances of the problem were used for comparison. The results are shown in Fig. 6. The MOPS algorithm gives similar or better packing solution using up to 30% fewer iterations on average.

6.3 Comparison on Circuit Layout Problems. In addition to 3D component layout problems, two 2D circuit layout problems involving printed circuit boards were also solved. PCBs are physical realizations of topological electric circuits. The physical design of PCBs involves the laying out of electrical packages such as integrated circuits (ICs), resistors, and capacitors, and routing the connections between them. The placement and routing processes are distinct activities. The actual routing is done either manually or by a routing tool after the placement. However, while placing the electrical packages the routability of the electrical connections between them needs to be taken into account.

The placement of electrical packages is an ideal candidate to be solved by our layout framework as a 2D problem. The packages are represented as 2D polygons and are required to be placed on the board which is itself a 2D polygon. Moreover, these 2D polygons are usually rectangles or combinations of rectangles. Constraints are in the form of fixed locations for packages, areas to keep inside of or remain outside of for packages and distance constraints between the packages.

We model the PCB layout task as a minimization problem where the objective function is the sum of pair wise overlaps between electrical packages and the protrusions of the packages outside the board. To this we also add a measure of proximity of circuit packages that are electrically connected to each other. By minimizing this objective function the pattern search algorithm (GPS or MOPS) places highly interconnected packages close to each other.

A 2D version of the sensitivity metric was used to drive the MOPS algorithm. The metric is exactly as described in Fig. 3. The patterns used were the translations along x and y axes of the packages.

The first problem involves the placement of approximately 250 packages on the board as shown in Fig. 7. The second problem instance involves placement of 36 major components of a circuit on the board as shown in Fig. 8.

The results in Figs. 7 and 8 show that the MOPS algorithm reduces run time by 30% on average. Also the MOPS algorithm is able to give better objective function values in the first example. The GPS algorithm could not obtain these objective function values by running the algorithm longer with increased number of iterations. This is because the GPS algorithm wastes the large moves of the small packages right in the beginning since it starts applying all patterns at the same time. Hence they are trapped inside bigger objects. In contrast the MOPS algorithm applies moves of the bigger packages first. When the placement of the bigger packages is almost decided, the patterns of the smaller packages start being applied. Now these packages can use their initial big moves to escape from local optima to reach a better solution.

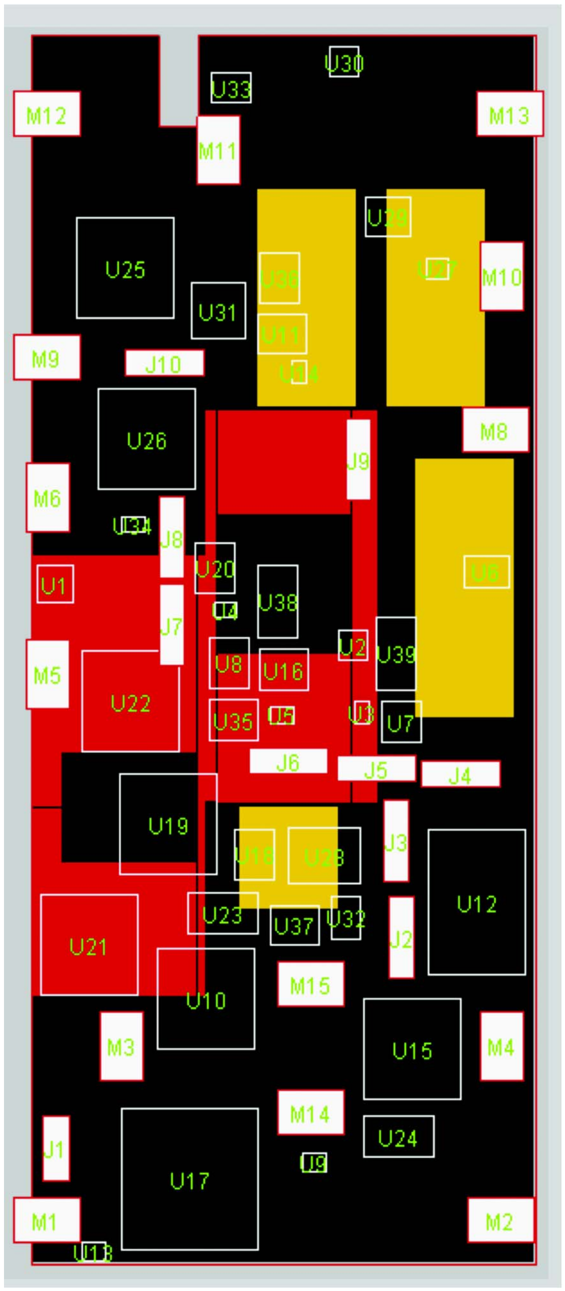
7 Discussion and Conclusions

We have introduced a computationally inexpensive metric called the sensitivity metric to estimate the effect of patterns on the objective function. This metric is used to schedule the patterns in the search to decrease runtime. Though this metric is specific to 2D and 3D component layout problems, we believe that similar metrics to schedule patterns can be derived from domain knowledge for other optimization problems where pattern search algorithms are used.

Results on three 3D component layout problems show that the MOPS algorithm performs 30% faster on average than the conventional GPS algorithm. The MOPS algorithm also improves the quality of the solution over the GPS algorithma, especially as seen in the circuit layout example. This improvement is not possible by simply increasing the number of iterations in the GPS algorithm. This is due to the fact that getting out of inferior local optima requires large step sizes and the large steps sizes are exhausted by the GPS algorithm earlier in the search without any success. Investigating the properties of search spaces which allow such behavior of the MOPS algorithm compared to the GPS algorithm is an interesting area for future work.

Acknowledgment

The authors would like to thank Dr. Jay McCormack for his comments on the paper and DesignAdvance Systems, Inc., for providing the circuit layout problem. This research effort was also partially sponsored by the Air Force Office of Scientific Research, Air Force Material Command, USAF, under Grant No. F045-019-0006. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of AFOSR or the U.S. Government.



🟥	Height restriction Region : Keepout U17 and U12
🟨	Height restriction Region : Keepout U17, U12, U21, U15, U10, U19, U22, U26 and U25
⬜	Fixed parts.
⬛	Printed Circuit Board.

RESULTS

GPS		MOPS		Imp (%)	
Obj.Fn.	#Iter	Obj.Fn.	#Iter	Obj.Fn.	#Iter
1.190	20,385	1.188	13,950	0.1	31.5
1.187	26,944	1.186	14,731	0.0	45.3
1.186	30,073	1.171	16,391	1.2	45.4
1.183	34,034	1.164	17,295	1.6	49.1

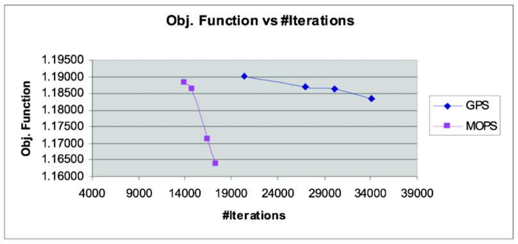


Fig. 8 Results comparing the GPS and the OPS algorithms for the layout of the second PCB board

References

[1] Yin, S., 2000, A Computational Framework for Automated Product Layout Synthesis Based on an Extended Pattern Search Algorithm, Ph.D. Thesis, Carnegie Mellon University, Pittsburgh, PA.
[2] Yin, S., and Cagan, J., 2000, "An Extended Pattern Search Algorithm for Three-Dimensional Component Layout," ASME J. Mech. Des., **122**(1), pp. 102–108.
[3] Ding, Q., and Cagan, J., 2003, "Automated Trunk Packing with Extended Pattern Search," Proceedings of the 2003 SAE Technical Conferences, Detroit, MI.
[4] Yin, S., Cagan, J., and Hodges, P., 2004, "Layout Optimization of Shapeable Components With Extended Pattern Search Applied to Transmission Design," J. Mech. Des., **126**(1), pp. 188–190.
[5] Torczon, V., and Trosset, M., 1997, "From Evolutionary Operation to Parallel Direct Search: Pattern Search Algorithms for Numerical Optimization," Comput. Sci. Statist., **29**(1), pp. 396–401.
[6] Aladahalli, C., Cagan, J., and Shimada, K., 2007, "Objective Function Effect Based Pattern Search—Theoretical Framework Inspired by 3D Component Layout," J. Mech. Des., **129**, March.
[7] Cagan, J., Shimada, K., and Yin, S., 2002, "A Survey of Computational Approaches to Three-Dimensional Layout Problems," CAD, **34**(8), pp. 597–611.
[8] Kawakami, T., Minagawa, M., and Kakazu, Y., 1991, "Auto Tuning of 3-D Packing Rules Using Genetic Algorithms," in Proceedings of the IEEE/RSJ International Workshop on Intelligent Robots and Systems IROS, Osaka, Japan.
[9] Corcoran, A. L., III, and Wainwright, R. L., 1992, "A Genetic Algorithm for Packing in Three Dimensions," Applied Computing: Technological Challenges of the 1990's—Proceedings of the 1992 ACM/SIGAPP Symposium on Applied Computing, Kansas City, KS.
[10] Wodziack, J., and Fadel, G. M., 1996, "Bi-Objective Optimization of Components Packing Using Genetic Algorithm," Proceedings NASA/AIAA/ISSMO Multidisciplinary Design and Optimization Conference, Seattle, WA.
[11] Ikonen, I., Biles, W., Kumar, A., Ragade, R. K., and Wissel, J. C., 1997, "A Genetic Algorithm for Packing Three-Dimensional Non-Convex Objects Having Cavities and Holes," Proceedings of 7th International Conference on Genetic Algorithms, Michigan State University, East Lansing, MI.
[12] Szykman, S., and Cagan, J., 1995, "A Simulated Annealing Approach to Three-Dimensional Component Packing," ASME J. Mech. Des., **117**(2A), pp.

308–314.
[13] Szykman, S., and Cagan, J., 1996, “Synthesis of Optimal Non-Orthogonal Routes,” ASME J. Mech. Des., **118**(3), pp. 419–424.
[14] Szykman, S., and Cagan, J., 1997, “Constrained Three Dimensional Component Layout Using Simulated Annealing,” ASME J. Mech. Des., **119**(1), pp. 28–35.
[15] Szykman, S., Cagan, J., and Weisser, P., 1998, “An Integrated Approach to Optimal Three Dimensional Layout and Routing,” ASME J. Mech. Des., **120**(3), pp. 510–512.
[16] Dickinson, J. K., and Knopf, G. K., 1998, “Serial Packing of Arbitrary 3D Objects for Optimizing Layered Manufacturing,” Proceedings SPIE Conference on Intelligent Robots and Computer Vision XVII: Algorithms, Techniques, and Active Vision, Boston, MA.
[17] Cagan, J., Degentesh, D., and Yin, S., 1998, “A Simulated Annealing-Based Algorithm Using Hierarchical Models for General Three-Dimensional Component Layout,” CAD, **30**(10), pp. 781–790.
[18] Aladahalli, C., Cagan, J., and Shimada, K., 2003, “Minimum Height Packing for Layered Manufacturing Using an Extended Pattern Search Algorithm,” Proceedings of the ASME DETC 2003, Chicago, IL.