Different Parallelizations of the Scatter Search Method

Van-Dat Cung[1], Thierry Mautor[1], Philippe Michelon[2]

Email: `Firstname.Name@prism.uvsq.fr`

1. Laboratoire PR*i*SM-CNRS URA 1525,
   Université de Versailles - Saint Quentin en Yvelines, France.
2. Département d'Informatique et Recherche Opérationelle,
   Université de Montréal, Canada.

Main Meta-heuristics for (Discrete) Optimization Problems

1. **Genetic Algorithm**
2. **Tabu Search**
3. Simulated Annealing
4. Grasp
5. ....

Comparison between Tabu and Genetic

- With Tabu, the search is based on the **objective function** while it is based on the **shape of good** solutions for Genetic Algorithms.
- Tabu takes into account the **history of the search** while a Genetic Algorithm **ignores** it.

Scatter Search

[F. Glover, 1977]

Main idea: Take advantage of Evolutionary Algorithms and Local Searches.

- Starts with a **collection of reference points**.
- A **trial point** is then created from a **weighted combination** of a small subset of reference points chosen among the best ones: **Elite set**. (Evolutionary).
- An **operator** is applied to the trial point (Local search).
- The resulting point replaces the worst reference point.
- The reference points used for the combination are prohibited for a while (**Tabu status**).
- Stops after a given number of iterations or when all the points are equal.

Scatter Search compared with Genetic Algorithms

Similarity : A set of feasible solutions evolves.

Main differences

- No metaphor with nature's behavior : linear combination of good solutions.
- Very first method combining more than two solutions.

Convenient : Easy to introduce advanced techniques (adaptive memory, intensification, diversification, etc.)

Drawback : Might be more difficult to analyze or to exploit.

Convenient :

For instance, combining points from subregions may either lead to subregions not explored yet if we could keep the collection of points as ”scattered” as possible (diversification), or search exhaustively one subregion if the points are from the same subregion.

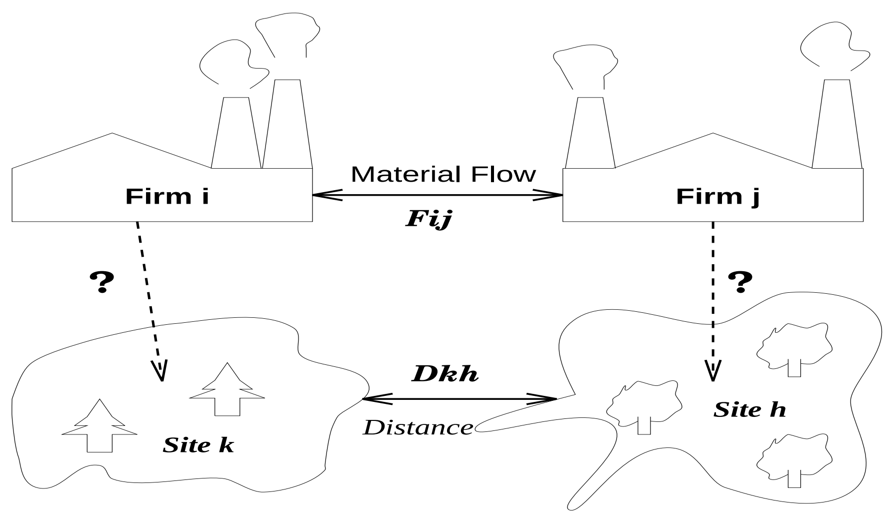
Drawback :

I.e. a trial point obtained by combining a large number of solutions may have no particular signification.

Quadratic Assignment Problem

[Koopmans & Beckmann, 1957]

How to assign n firms to n sites in order to minimize the cost, which depends on the **flows** F between the firms and the **distances** D between the sites ?



The problem is **NP-complete** (TSP) and the size of the solution space is $n!$.

Other applications : VLSI, architecture, hospital services, ergonomic design, etc.

TSP is a special case of QAP —> QAP is NP-complete and $n!$ for its solution space.

To show that, it just needs to use a special F matrix while the matrix D is the distances between the cities in the TSP (cf. Mautor's PhD 93).

Even to find an ϵ-approximation is NP-complete.

Mathematical Formulation

$$\begin{aligned}
\min \quad & \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n}\sum_{h=1}^{n} F_{ij} D_{kh} x_{ik} x_{jh} \\
\text{s.t.} \quad & \\
\sum_{i=1}^{n} x_{ik} &= 1, \qquad k \in \{1,2,...,n\}, \ \textbf{one firm/site} \\
\sum_{k=1}^{n} x_{ik} &= 1, \qquad i \in \{1,2,...,n\}, \ \textbf{one site/firm} \\
x_{ik} &= 0 \text{ or } 1, \qquad i,k \in \{1,2,...,n\},
\end{aligned}$$

$$\begin{aligned}
F_{ij} &= \text{flow of material between firm } i \text{ and firm } j, \\
D_{kh} &= \text{distance between site } k \text{ and site } h, \\
x_{ik} &= 1 \text{ if firm } i \text{ is assigned to site } k, \text{ 0 otherwise.}
\end{aligned}$$

In shorter term :

$$\min \sum_{i=1}^{n}\sum_{j=1}^{n} F_{ij} D_{p(i)p(j)}$$

where p is a permutation of the set $\{1,2,...,n\}$.

Thus, a solution is a vector of length n.

Generating the Reference Set

The reference set is an important aspect of evolutionary algorithms which is **generally overlooked** and leads to **hazardous** generation.

A "good" reference set for QAP should have at least two main characteristics :

1. All the possibilities ("possible genes") should be contained in it.

   For QAP, it means that there should be a solution for each firm k located on eac[h] site i.

2. As the best subregions is unknown, the reference set should cover as good as possible all the feasible solution space.

   A priori, this could be achieved by generating a reference set which maximizes the euclidean minimum distance between any couple of points.

   For the QAP, the maximum distance between any two points is $\sqrt{2n}$.

Circular Permutation for the Reference Set (1/4)

To satisfy these criteria :

1. We first generate at random a feasible solution x^1.
2. Let now p be a circular permutation of $\{1, ..., n\}$.
3. A second solution x^2 is constructed by assigning $p(k_i^1)$ to site i, where k_i^1 is the firm located on the site i in x^1.

   A third solution x^3 is then constructed by assigning $p(k_i^2)$ to site i, where k_i^2 is the firm located on the site i in x^2, etc.

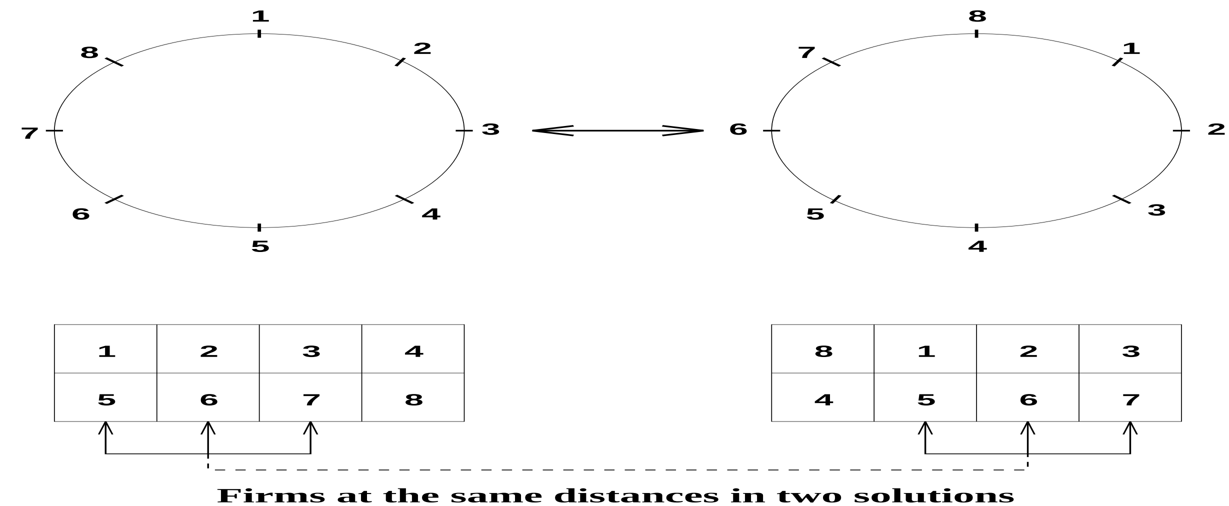
The process is repeated $n - 1$ times to have $|E| = n$.

If more reference points are needed, a second feasible solution is drawn and permuted Finally, Tabu Search is applied on them for improvement.

In practice, we generate $2n$ improved reference points.

It can be easily shown that these n solutions satisfy the criteria.

Circular Permutation for the Reference Set (2/4)



In order to break the symmetric solutions obtained by permutation, we applied a more complex permutation which prohibits, whenever is possible, for a couple of firm to be located at the same distance in two different solutions.

Circular Permutation for the Reference Set (3/4)

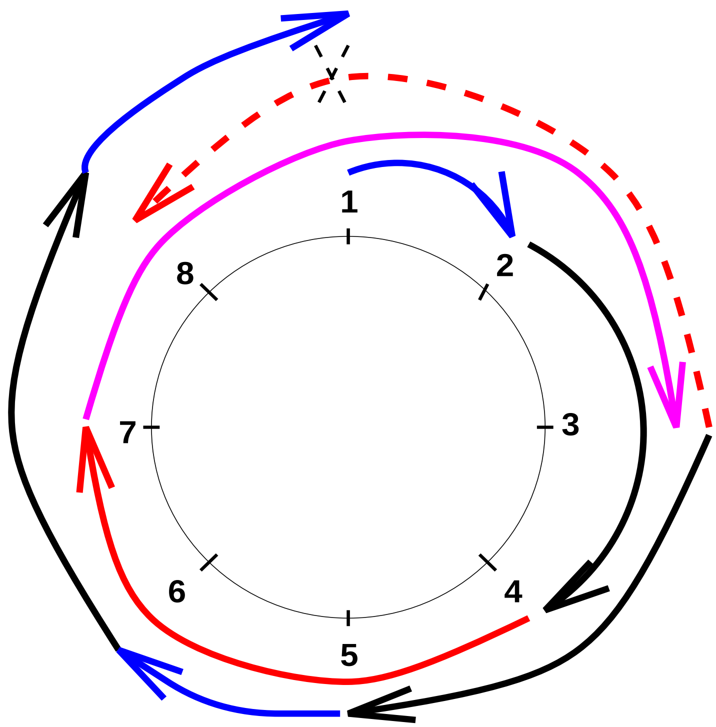
TabuList=$\emptyset$; $i = 1$; Set site 1 as used;

REPEAT $n - 1$ TIMES

- Search first $p - i$ such that $p - i$ is not used and $d(i, p - i)$ is not tab
- IF $\nexists(p - i)$ THEN
  - TabuList=$\emptyset$ /* Initialize again */
  - Search first $p - i$ such that $p - i$ is not used
- $p(i) = p - i$
- Set $p - i$ as used
- Insert $d(i, p - i)$ in TabuList
- $i = p - i$

$p(1) = i$

Circular Permutation for the Reference Set (4/4)



1	2	3	4	5	6	7	8
8	1	7	2	3	5	4	6
6	8	4	1	7	3	2	5
5	6	2	8	4	7	1	3
3	5	1	6	2	4	8	7
7	3	8	5	1	2	6	4
4	7	6	3	8	1	5	2
2	4	5	7	6	8	3	1
1	2	3	4	5	6	7	8

P1	P2	P3	P4	P5	P6	P7	P8
2	4	5	7	6	8	3	1

Combining Solutions

Elitist selection : a Elite subset E of feasible solutions is maintained.

A number b, uniformly drawn between 2 and a parameter B, of solutions $\{x^1, x^2, ..., x^b\}$ are then chosen from E for combination.

In practice, $B \simeq 5$ and $|E| \simeq 40\% \times n$.

The roulette wheel has been tried and has given solutions slightly worse (useless diversification).

Linear combination : a solution $x = \sum_{i=1}^{r} \lambda_i x^i$ where $\lambda_i \in \mathbb{B}$ **may not be feasible**.

In practice, $\lambda_i = 1$ (fairness between "parents").

Find a Feasible Solution Closest to x

We have to solve the problem.

$$
\begin{aligned}
\min \quad & ||\hat{x} - x||^2 & (1) \\
s.t. \quad & \sum_{i=1}^{n} \hat{x}_{ik} = 1, \quad k \in \{1, ..., n\}, & (2) \\
& \sum_{k=1}^{n} \hat{x}_{ik} = 1, \quad i \in \{1, ..., n\}, & (3) \\
& \hat{x}_{ik} = 0 \text{ or } 1 \quad i, k \in \{1, ..., n\}, & (4)
\end{aligned}
$$

where $||\hat{x} - x||$ is the euclidean distance between $\hat{x}$ and x.

This is equivalent to solve:

$$\max \sum_{i=1}^{n} \sum_{k=1}^{n} x_{ik} \hat{x}_{ik}$$

$$\text{subject to } (2), (3), (4),$$

because $\sum_{i=1}^{n} \sum_{k=1}^{n} \hat{x}_{ik}^2 = n$,
and $\sum_{i=1}^{n} \sum_{k=1}^{n} x_{ik}^2$ does not depend on $\hat{X}$.

This could be solved in $O(n^3)$. But ...

But ... we preferred a simpler construction.

"Not-so-greedy" Heuristic in $O(n^2)$

`AvailableSites` = $\{1, ..., n\}$

`AvailableFirms` = $\{1, ..., n\}$

$\hat{x}_{ik} = 0, \quad i = 1, ..., n \quad k = 1, ..., n$

`FOR` $t = 1, n$ `DO`

- `Choose at random (according to a uniform distribution) a site` $i \in$ `AvailableSites`
- `Choose at random (according to a uniform distribution) a firm` $k \in \texttt{Argmax}\{x_{ik} | k \in \texttt{AvailableFirms}\}$
- $\hat{x}_{ik} = 1$
- `AvailableSites = AvailableSites-`$\{i\}$
- `AvailableFirms = AvailableFirms-`$\{k\}$

Convenient : easy to introduce diversification (random process+frequency).

Other heuristics have been tested without any significant change in the numerical results.

Improving Operator

Aim : Generate one or *several* improved solution(s).

How ? Basic Tabu Search for a fixed number of iterations I_{TS1}.

A move : Exchange the firms located on sites i_1 and i_2.

Neighborhood : A point which can be reached in the set of solutions from an other in one move (best move or *chosen randomly between the 10 bests moves*).

(Unrestrictive) Tabu list :

- Once, firm k_1 is assigned on i_2 and firm k_2 on i_1, the reverse move (k_1 on i_1 and k_2 on i_2) is forbidden.
- The size T of the Tabu list is a parameter.
  In practice, T_{TS} has no effect on small size problems ($n \leq 90$) while T_{TS} depends on the data for large size problems.

Insertion strategy : The improved solution is inserted into the reference set only if its value is better than the worst (the closest has been tested without success).

New idea : *A new idea consists in generating BC improved solutions instead of only one.* This is done to keep the "search effort" spent in the Tabu Search process (intensification).

Tabu move :

Even k_1 on i_1 and k_2 on i_2 is forbidden, it is still possible to exchange firms on i_1 and i_2 if the firms differ from k_1 or k_2.

Insertion strategy : And its structure is not equal to a existing solution already in the reference set.

In an earlier version, the new solution is not compared for insertion with the **worst** but with the **closest** (in particular, it ensures of not generating a point which is yet in the population). This is done in order to keep the diversity of the reference set. But the numerical results are slightly worse, because this strategy has the drawback of suppressing points which might be good in term of cost.

By the way, the diversification could be achieved otherwise.

Adding Diversification and Intensification

Keep the reference set as scattered as possible :

1. To avoid some good solutions to be too often used for combination, a status Tabu T_{ScS} for a certain number of iterations is assigned to a solution used fo combination.
2. Use a matrix frequency $F = (F_{ik})$, updated at each iteration of Scatter Searc $(F = F + X)$, to report the number of times where firm k was assigned to site i in the starting points of the Tabu Search (improving operator). Then in "diversification" iterations N_D, a new solution is created with $\lceil C_D \times n \rceil$ (C_D is a parameter in [0,1]) assignments of firms to sites which wer **not frequently** used.

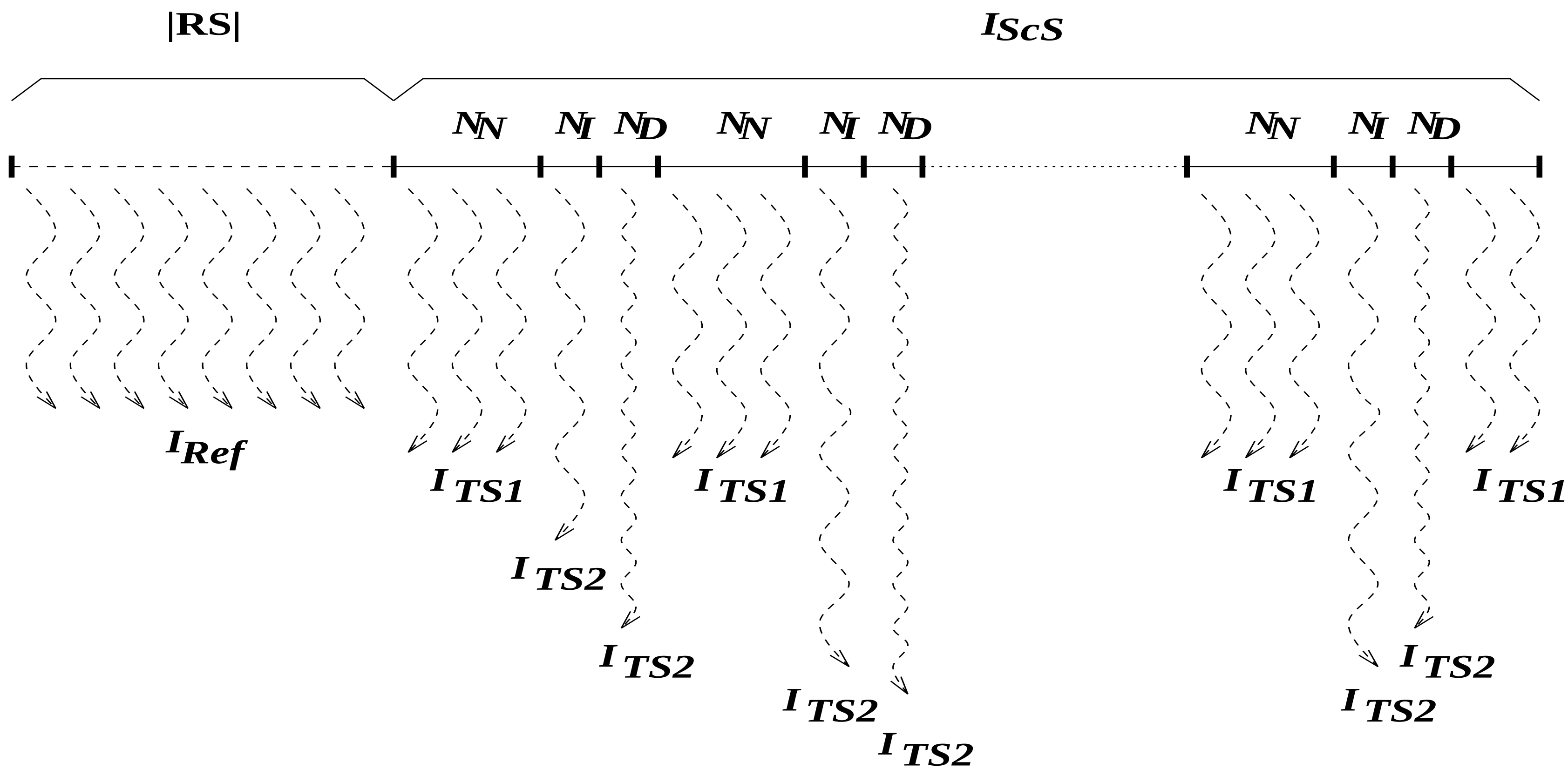
(Inten/diver)sification : Just iterate more Tabu Search, N_I and I_{TS2}, respectively N_D and I_{TS2}. *I_{TS2} is now the number of Tabu Search iterations without improvement instead of just a fixed number of iterations.*

The algorithm uses for diversification of the $\lceil cn \rceil$ firm is the same as the combining procedure, but with the F matrix instead of T, and Argmin instead of Argmax.

`AvailableSites =` $\{1, ..., n\}$
`AvailableFirms =` $\{1, ..., n\}$
$\hat{x}_{ik} = 0, \quad i = 1, ..., n \quad k = 1, ..., n$
`FOR` $t = 1, \lceil cn \rceil$ `DO`

- `Choose at random (according to a uniform distribution) a site` $i \in$ `AvailableSites`
- `Choose at random (according to a uniform distribution) a firm` $k \in \texttt{Argmin}\{F_{ik} | k \in \texttt{AvailableFirms}\}$
- $\hat{x}_{ik} = 1$
- `AvailableSites = AvailableSites-`$\{i\}$
- `AvailableFirms = AvailableFirms-`$\{k\}$

Scatter Search process



Numerical Results on QAPLIB

[Burkard, Karisch & Rendl, 1991]

Problems		1	50	100	500	1000	2500
Els19	Avg.	17356429	17212548	17212548	17212548	17212548	17212548
=	Best	17212548	17212548	17212548	17212548	17212548	17212548
Bur26a	Avg.	5431050	5427604	5426842	5426670	5426670	5426670
=	Best	5426670	5426670	5426670	5426670	5426670	5426670
Kra30a	Avg.	89769	89301	89152	88900	88900	88900
=	Best	88900	88900	88900	88900	88900	88900
Lipa90b	Avg.	15081175	13502112	12994369	12490441	12490441	12490441
=	Best	15053917	12490441	12490441	12490441	12490441	12490441

Numerical Results on QAPLIB

Problems		1	50	100	500	1000
Sko100a	Avg.	155055	153608	153292	152654	152242
=	Best	152834	152324	152324	152184	152002
Tai100a (0.099%)	Avg.	22377260	21322968	21312543	21273103	21263123
–21125314	Best	21527328	21236086	21236086	21203868	21168282
Tho150	Avg.	8731827	8200765	8193156	8157914	8147082
+8134030	Best	8240638	8168434	8160642	8139894	8136140
Tai150b (0.024%)	Avg.	518907000	506570000	504799000	500863000	500227000
–499348972	Best	504014559	504014559	501165936	499560242	<u>499468095</u>
Tai256c	Avg.	45207903	44865244	44851341	44828651	
+44894480	Best	**44824542**	**44824542**	**44823712**	**44822924**	

Numerical Results on QAPLIB (small/medium sizes)

[Burkard, Karisch & Rendl, 1991]

Pbs.	Best in QAPLIB (06/27/1997)	Avg. solution I_{ScS}=2500	Gap1 (LB-Avg.)/LB	Best found over all	Gap2 (LB-UB)/LB
els19	17212548	17212548	0.000%	17212548	0.000%
nug20	2570	2570	0.000%	2570	0.000%
bur26a	5426670	5426670	0.000%	5426670	0.000%
kra30a	88900	88900	0.000%	88900	0.000%
sko56	34458	34458	0.000%	34458	0.000%
tai80b	818415043	821216869	+0.340%	818415043	0.000%
lipa90b	12490441	12490441	0.000%	12490441	0.000%

Numerical Results on QAPLIB (large sizes)

Pbs.	Best in QAPLIB (06/27/1997)	Avg. solution I_{ScS}=2500	Gap1 (LB-Avg.)/LB	Best found over all	Gap2 (LB-UB)/LB
`sko100a`	152002	152087	+0.056%	152002	0.000%
`tai100a`	21125314	21162417	+0.018%	<u>21133392</u>	+0.038%
`wil100`	273038	273047	+0.003%	273038	0.000%
`tai150b`	499342577	499942142	+0.120%	<u>499450869</u>	+0.022%
`tho150`	8134030	8140154	+0.075%	**8133864**	-0.002%
`tai256c`	44787190	44826698	+0.088%	<u>44821704</u>	+0.077%
R. N.		44775849	-0.025%	**44759294**	-0.062%

Details of the ten best solutions

# best Sol. I_{ScS}=2500	sko100a	tai100a	wil100	tai150b	tho150	tai256c	tai256c R. N.
1	152002	21133392	273038	499468095	8133864	44821704	44759294
2	152014	21146176	273038	499495974	8134406	44822924	44759294
3	152042	21160418	273044	499537617	8136378	44823712	44765184
4	152044	21162658	273044	499710131	8138156	44824542	44779612
5	152098	21163412	273044	499787244	8138160	44827964	44779770
6	152102	21168288	273046	500018593	8139412	44828190	44780610
7	152104	21168494	273054	500149901	8141208	44828230	44783014
8	152128	21169048	273054	500332735	8145404	44829548	44783014
9	152160	21170818	273054	500456593	8146598	44829938	44784118
10	152184	21181474	273054	500464538	8147956	44830236	44784588

Convergences of the Reference Set

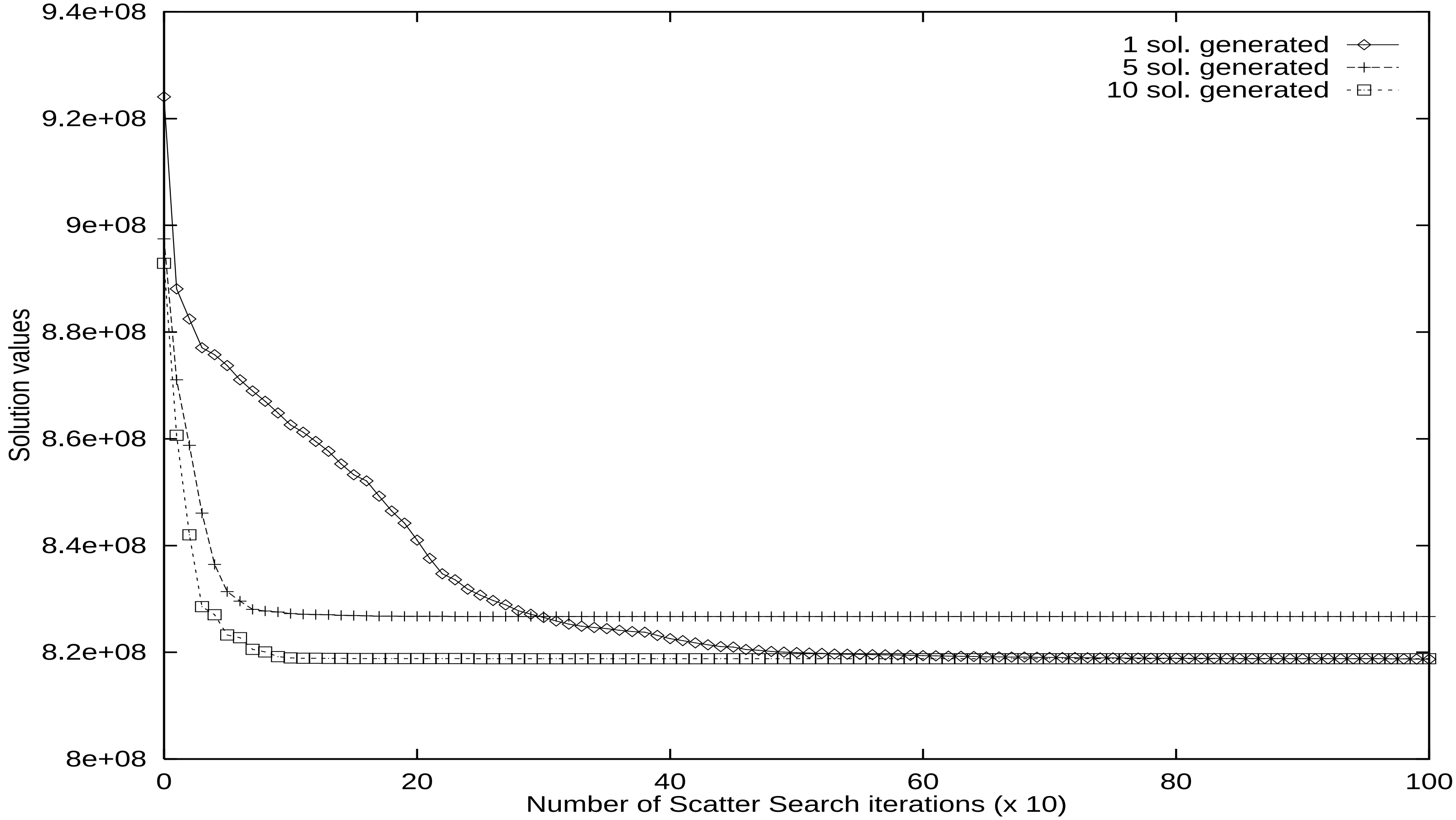


Figure 1: Worse solutions obtained on `tai80b`.

Convergences of the Reference Set

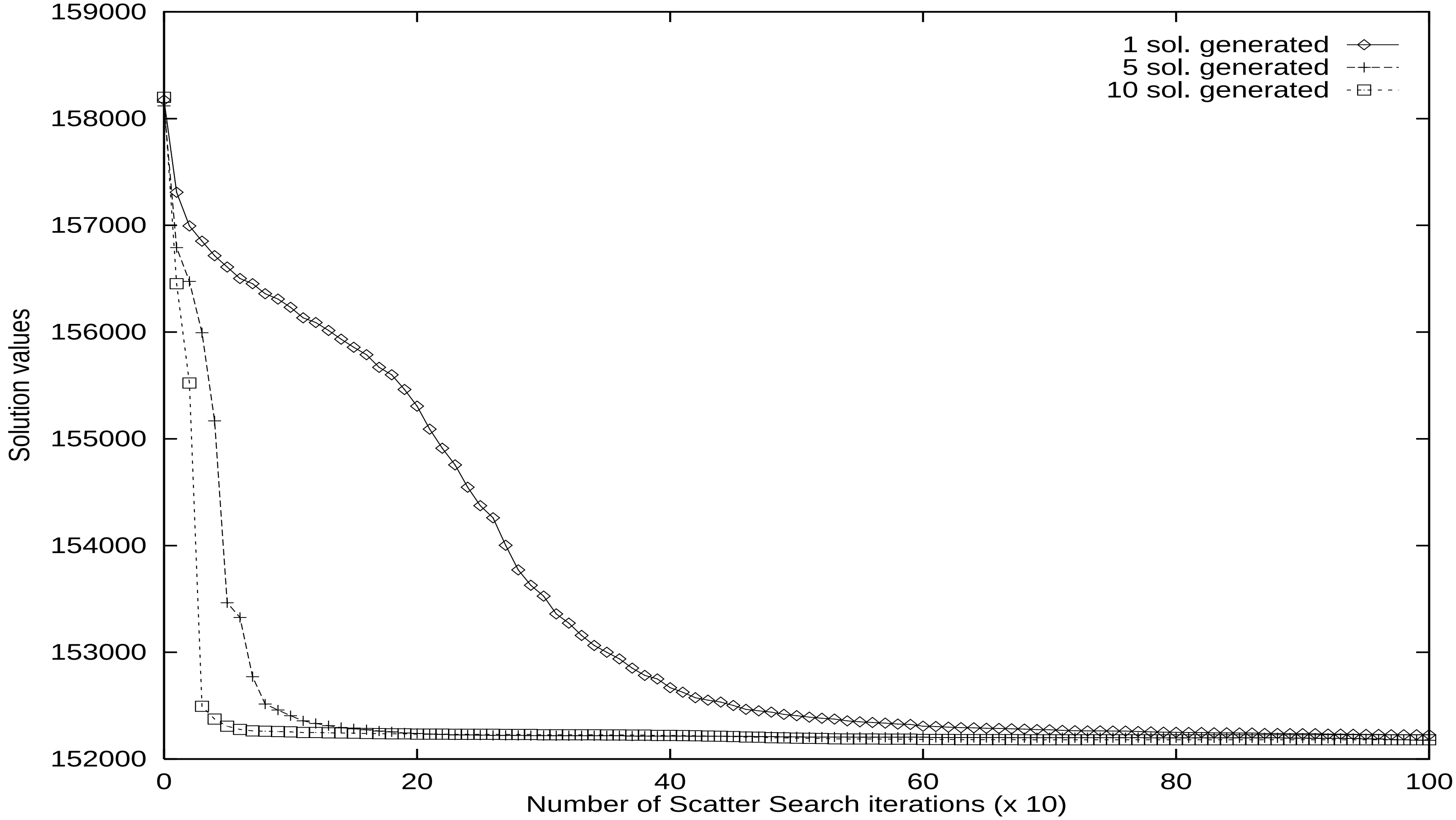


Figure 2: Worse solutions obtained on `sko100a`.

Concluding Remark and On-going Works

Scatter Search seems to be an **extremely competitive** metaheuristic method for the QAP.

Reference Set Instead of generating randomly a solution, we are working a greedy algorithm.

Improving operator

- Mimausa.
- Self-tuning the size of the Tabu list.
- More restricted neighborhood (from time to time).

Insertion strategy

- Compared to the worst even it is Tabu, not only in the reference set.
- Add the last point obtained on Diversification et Intensification iterations.

Stop condition I_T iterations without improvement.

Parallelization of the Scatter Search, using clustering technique.