On the Effect of Connectedness for Biobjective Multiple and Long Path Problems

Sébastien Verel[1,3] Arnaud Liefooghe[2,3] Jérémie Humeau[3,4]
Laetitia Jourdan[3] Clarisse Dhaenens[2,3]

[1] University of Nice Sophia Antipolis – CNRS, France

[2] Université Lille 1, LIFL – CNRS, France

[3] INRIA Lille-Nord Europe, France

[4] IA department, École des Mines de Douai, France

DOLPHIN Team, SCoBI Team

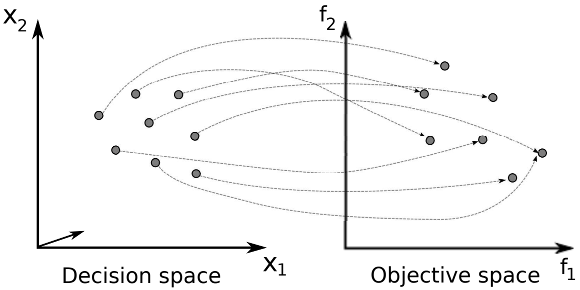
sebastien.verel@inria.fr

Rome, January 20, 2011

Multiobjective combinatorial optimization

Multiobjective combinatorial optimization (MoCO) problem

- $\mathcal{X}$: enumerable set of feasible solutions in the *decision space*
- $M \geq 2$ objective functions $(f_1, f_2, \ldots, f_M)$
- $\mathcal{Z} = f(\mathcal{X}) \subseteq \mathbb{R}^M$: set of feasible outcome vectors in the *objective space*

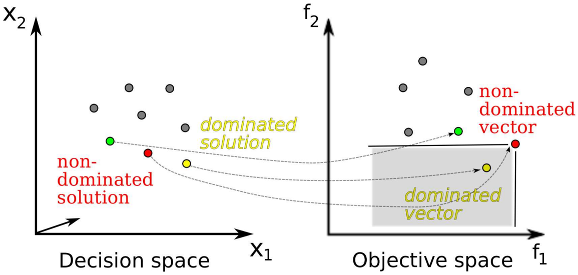


Multiobjective combinatorial optimization

Pareto dominance between solutions (maximization)

A solution $x \in \mathcal{X}$ dominates a solution $x' \in \mathcal{X}$ iff

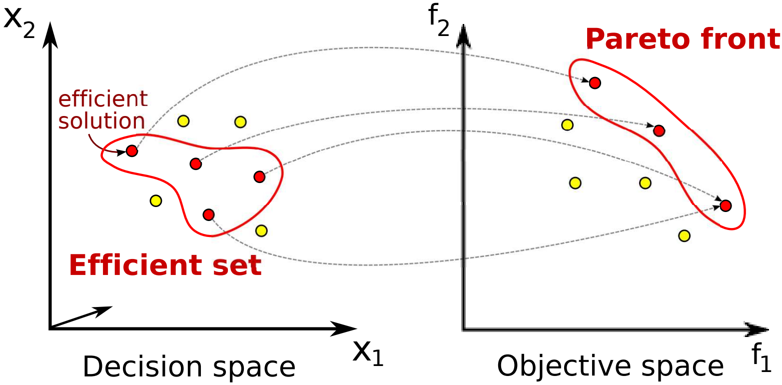
- $\forall i \in \{1, 2, \ldots, M\}$, $f_i(x') \leq f_i(x)$
- $\exists j \in \{1, 2, \ldots, M\}$ such that $f_j(x') < f_j(x)$



Efficient set (or non-dominated set or Pareto optimal set)

Efficient set (or non-dominated set or Pareto optimal set)

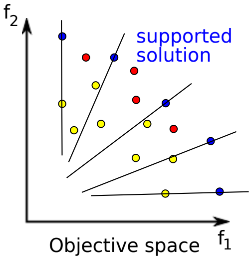
Set of non-dominated solutions

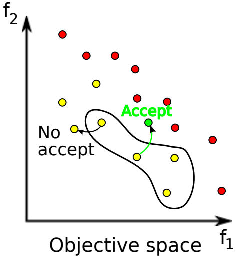


How to solve MoCO problems with a local search algorithm ?

Two main classes :

- Scalar approaches : multiple scalarized aggregations of the objective functions, only able to find a subset of efficient solutions (supported solutions)
- Pareto-based approaches : directly or indirectly focus the search on the Pareto dominance relation

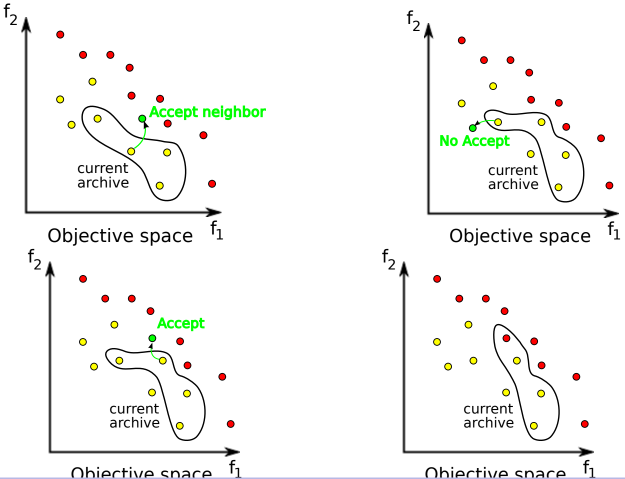




Example of Pareto-based approach

Pareto Local Search [Paquete et al., 2004]

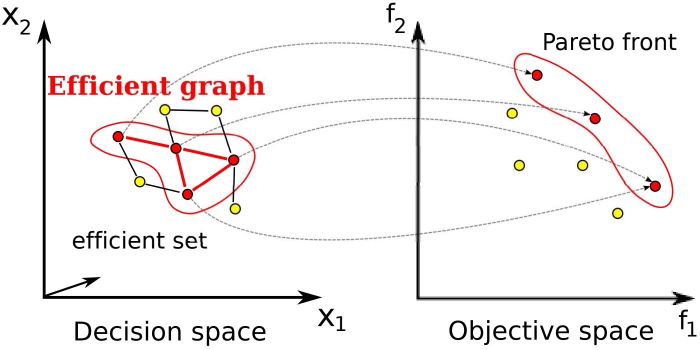
- Archive of mutually non-dominated solutions
- Iteratively improve this archive by exploring the neighborhood



Connectedness

Efficient graph [Ehrgott and Klamroth, 1997]

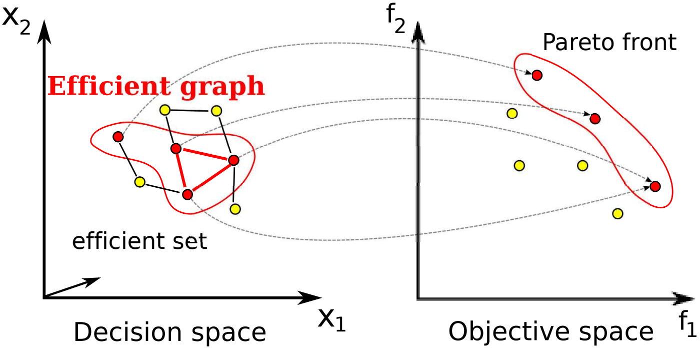
- $G = (V, E)$ where V consists of all efficient solutions and E connects two solutions if they are neighbors
- Efficient set is connected if G is connected



Connectedness

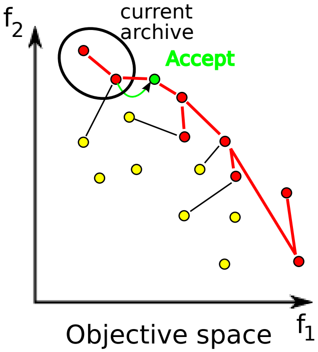
Efficient graph [Ehrgott and Klamroth, 1997]

- $G = (V, E)$ where V consists of all efficient solutions and E connects two solutions if they are neighbors
- Efficient set is connected if G is connected



Two-phase approaches when connectedness is hold

- From an efficient solution (find with a scalar approach for example)
- Iteratively find a new efficient solution by exploring the neighborhood (Pareto-based approach)



Claims

- Connectedness : strong motivation on the LS design
- Structure of this efficient set : crucial role for LS design [Gorski et al., 2006]

Main difficulties in MoCO

Why identifying the efficient set is **difficult** [Ehrgott, 2005] ?

- NP-complete : deciding if a solution is efficient is NP-complete for most MoCO problems (even if no single-objective counterpart is NP-hard)
- Intractability : number of efficient solutions (non-dominated vectors) typically exponential in the problem size

⇒ Identify a "good" efficient set **approximation**, such that :

(*i*) close to Pareto front

(*ii*) well-spread along the Pareto front

Intractability : could be catastrophic for two-phase approaches, or Pareto-based approaches when the goal is not to enumerate the whole efficient set...

Motivations of this work

The connectedness is not the holy grail

We would like to show that :

When the MoCO problem is intractable :

- Connectedness is not relevant to explain the difficulty, and to improve the design of LS

Motivations of this work

The connectedness is not the holy grail

We would like to show that :

When the MoCO problem is intractable :

- Connectedness is not relevant to explain the difficulty, and to improve the design of LS

Even more...

- Reconsiderer the relevant characteristics of MoCO problems which can explain the dynamics of algorithms, difficulty of instances, help to design more efficient methods

Long k-Path problems : an old story in mono-objective

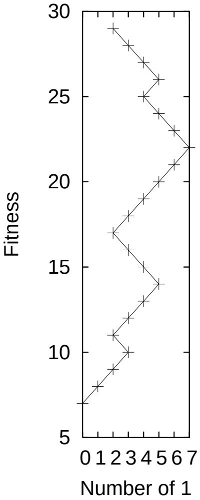
Goal [Horn et al., 1994]

Unimodal problem is not always easy :
path length to global optimum is also a criterium of difficulty

Properties of long k-path

- Unimodal problem :
  - Path length grows exp. with bitstring size
  - Hamming distance between two consecutive solutions is 1
  - Any other solution at distance 1 is off the path
- Very long path to global optimum
- "Short-cuts" require at least k bit-flips

Long 2-path $l = 7$



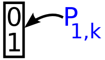
Results on long k-Path problems

Length

$|P_{l,k}| = (k+1)2^{(l-1)/k} - k + 1$ is the length of the k-path of dimension l.

Expected running time

- For a hillclimbing algorithm
  chooses the best solution in the neighborhood defined by Hamming distance 1 :
  $l \cdot |P_{l,k}|$
- For a $(1+1)$ EA
  flips each bit with a probability $p = 1/l$ at each iteration :
  $\mathcal{O}(l^{k+1}/k)$ [Rudolph, 1996]



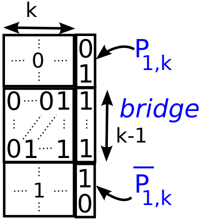
Results on long k-Path problems

Length

$|P_{l,k}| = (k+1)2^{(l-1)/k} - k + 1$ is the length of the k-path of dimension l.

Expected running time

- For a hillclimbing algorithm
  chooses the best solution in the neighborhood defined by Hamming distance 1 :
  $l \cdot |P_{l,k}|$
- For a $(1+1)$ EA
  flips each bit with a probability $p = 1/l$ at each iteration :
  $\mathcal{O}(l^{k+1}/k)$ [Rudolph, 1996]



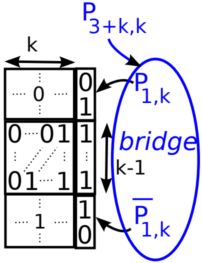
Results on long k-Path problems

Length

$|P_{l,k}| = (k+1)2^{(l-1)/k} - k + 1$ is the length of the k-path of dimension l.

Expected running time

- For a hillclimbing algorithm
  chooses the best solution in the neighborhood defined by Hamming distance 1 :
  $l \cdot |P_{l,k}|$
- For a $(1+1)$ EA
  flips each bit with a probability $p = 1/l$ at each iteration :
  $\mathcal{O}(l^{k+1}/k)$ [Rudolph, 1996]



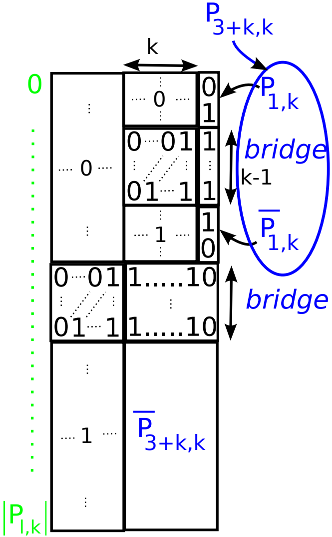
Results on long k-Path problems

Length

$|P_{l,k}| = (k+1)2^{(l-1)/k} - k + 1$ is the length of the k-path of dimension l.

Expected running time

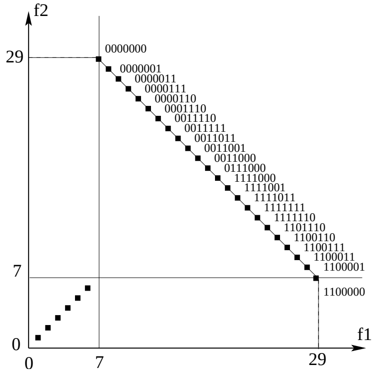
- For a hillclimbing algorithm chooses the best solution in the neighborhood defined by Hamming distance 1 : $l \cdot |P_{l,k}|$
- For a $(1+1)$ EA flips each bit with a probability $p = 1/l$ at each iteration : $\mathcal{O}(l^{k+1}/k)$ [Rudolph, 1996]



Bi-objective Long k-Path problems

Connected bi-objective problem : more difficult for multiobjective hillclimbing (PLS) than simple multiobjective EA

Long 2-Path with $l = 7$



Definition

Efficient set = long path $P_{l,k}$

Properties

- Intractable efficient set : $\Rightarrow$ eff. set approximation
- Running time of the PLS : exponential,
- Simple EA can take shortcuts

Multiobjective LS *vs.* EA

Pareto Local Search (PLS)

1. One solution is chosen at random from the archive.
2. *All solutions located at Hamming distance* 1 *are evaluated*
3. Check for insertion in the archive
4. The current solution is then marked as *visited*

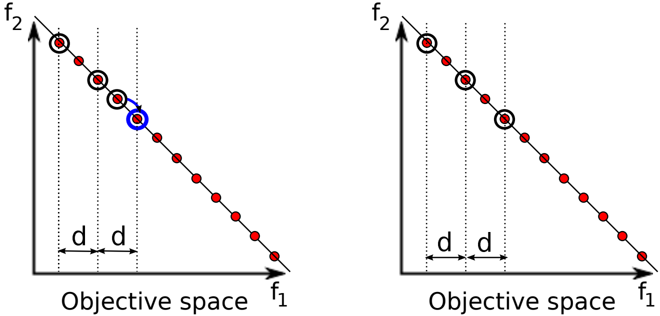
Simple Evolutionary Multiobj. Optimization (SEMO)

1. One solution is randomly chosen from the archive
2. *Each bit of this solution is independently flipped with a probability* $p = 1/l$
3. Check for insertion in the archive

All solutions are potentially reachable

- Starting from on the first solution of the long k-path
- Stop when 98% of the optimal hypervolume for archive of size 100 is reached

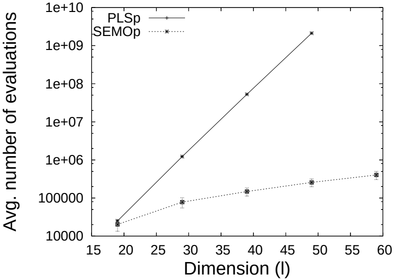
Limited size archive : "ideal" archive update



Archiving method

- Limited-size archive
- Reduce archive size such that $f_1(a_{i+1}) - f_1(a_i) \geq d$

Experimental results



Results

- Pareto LS : Number of evaluations grows exponentially with problem dimension
- Simple Multiobjective EA outperforms Pareto LS

First conclusions

- From the point of view of Pareto LS :
  - Efficient graph is linear,
  - Distance between efficient solutions is large
- From the point of view of SEMO :
  - Efficient graph : fully connected, high degree
  - Distance between efficient solutions is short

First conclusions

- From the point of view of Pareto LS :
  - Efficient graph is linear,
  - Distance between efficient solutions is large
- From the point of view of SEMO :
  - Efficient graph : fully connected, high degree
  - Distance between efficient solutions is short

First conclusions on difficulty

Connectedness
+ Structure of the efficient graph !

First conclusions

- From the point of view of Pareto LS :
  - Efficient graph is linear,
  - Distance between efficient solutions is large
- From the point of view of SEMO :
  - Efficient graph : fully connected, high degree
  - Distance between efficient solutions is short

First conclusions on difficulty

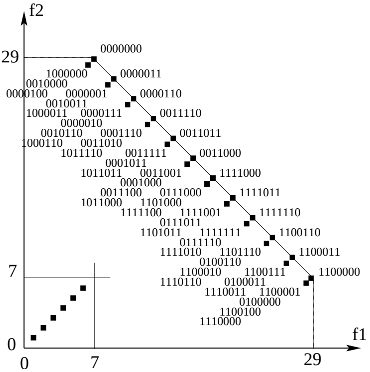
Connectedness
+ Structure of the efficient graph !

But it is still not enought...

Biobjective Multiple k-Path problems

Disconnected bi-objective problem : more difficult for simple multiobjective EA than multiobjective hillclimbing (PLS)

Multiple 2-Path with $l = 7$

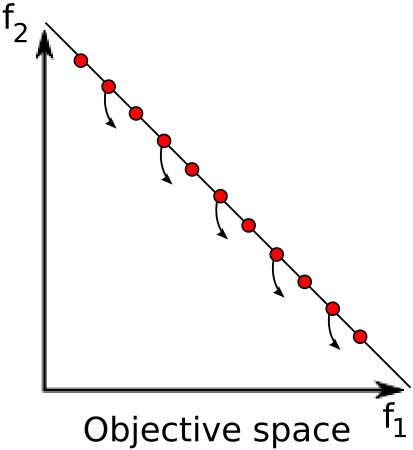


Definition

Modify the previous biobjective long k-Path :

- Disconnect the efficient graph
- Add some "dominate" paths between solutions

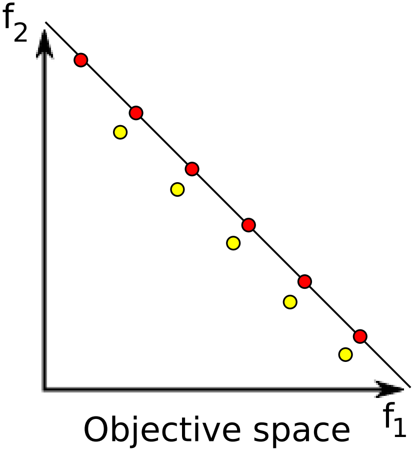
Biobjective Multiple k-Path problems



- Disconnect the efficient graph : move 1 over 2 solutions "under" the next solution in the path

example of bridges with $k = 3$

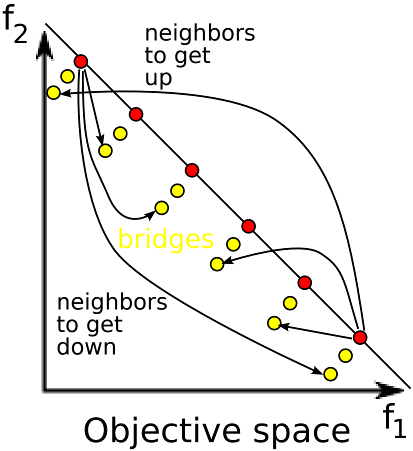
Biobjective Multiple k-Path problems



- Disconnect the efficient graph :
  move 1 over 2 solutions "under" the next solution in the path

example of bridges with $k = 3$

Biobjective Multiple k-Path problems



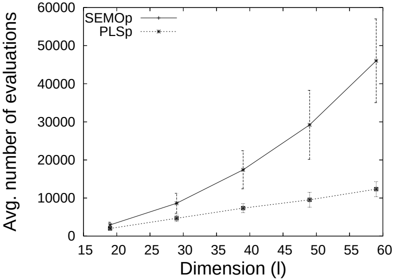
- Add recursively "dominate" paths between solutions :
  Add some bridges recursively

example of bridges with $k = 3$

$u0^k P(i_0)$	??????000.111.0
bridge	??????001.111.0
down	??????011.111.0
$u1^k P(i_1)$	??????111.111.0

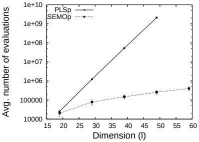
$u0^k P(i_0)$	??????000.111.0
bridge	??????100.111.0
up	??????110.111.0
$u1^k P(i_1)$	??????111.111.0

Experimental results

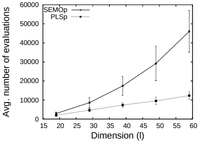


- Multiobjective LS : number of evaluations grows nearly linearly with problem dimension
- Pareto LS outperforms Simple Multiobjective EA

Summary on results



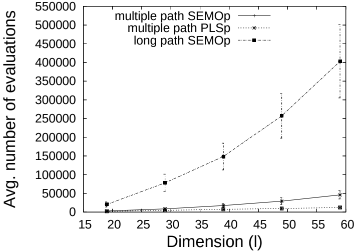
Long 2-path problem



Multiple 2-path problem

PLS <<< SEMO

PLS >> SEMO



long PLS <<< long SEMO <<< multiple SEMO << multiple PLS

Conclusions and future works

- Connectedness, and structure of efficient set is not the key concept to explain the difficulty

Reconsider relevant features of multiobjective problems

- Search space of multiobjective problems is the set of sets

$$\Sigma = \{\sigma \subset S \ : \ |\sigma| \leq n\}$$

- Dynamics of LS, and difficulty of problem instance could be explain in Σ (prob. to improve a set, local optimum sets, etc.)

Future works

- Running time analysis for the biobjective 'path' problems
- Study the fitness landscape (and dynamics of LS) where the search space is Σ

Bibliography

Ehrgott, M. (2005).
Multicriteria optimization.
Springer, second edition.

Ehrgott, M. and Klamroth, K. (1997).
Connectedness of efficient solutions in multiple criteria combinatorial optimization.
European Journal of Operational Research, 97(1) :159–166.

Gorski, J., Klamroth, K., and Ruzika, S. (2006).
Connectedness of efficient solutions in multiple objective combinatorial optimization.
Technical Report 102/2006, University of Kaiserslautern, Department of Mathematics.

Horn, J., Goldberg, D. E., and Deb, K. (1994).
Long path problems.
In *PPSN*, volume 866 of *LNCS*, pages 149–158. Springer.

Paquete, L., Chiarandini, M., and Stützle, T. (2004).
Pareto local optimum sets in the biobjective traveling salesman problem : An experimental study.
In *Metaheuristics for Multiobjective Optimisation*, volume 535 of *Lecture Notes in Economics and Mathematical Systems*, chapter 7, pages 177–199. Springer.

Rudolph, G. (1996).
How mutation and selection solve long path problems in polynomial expected time.
Evolutionary Computation, 4(2) :195–205.