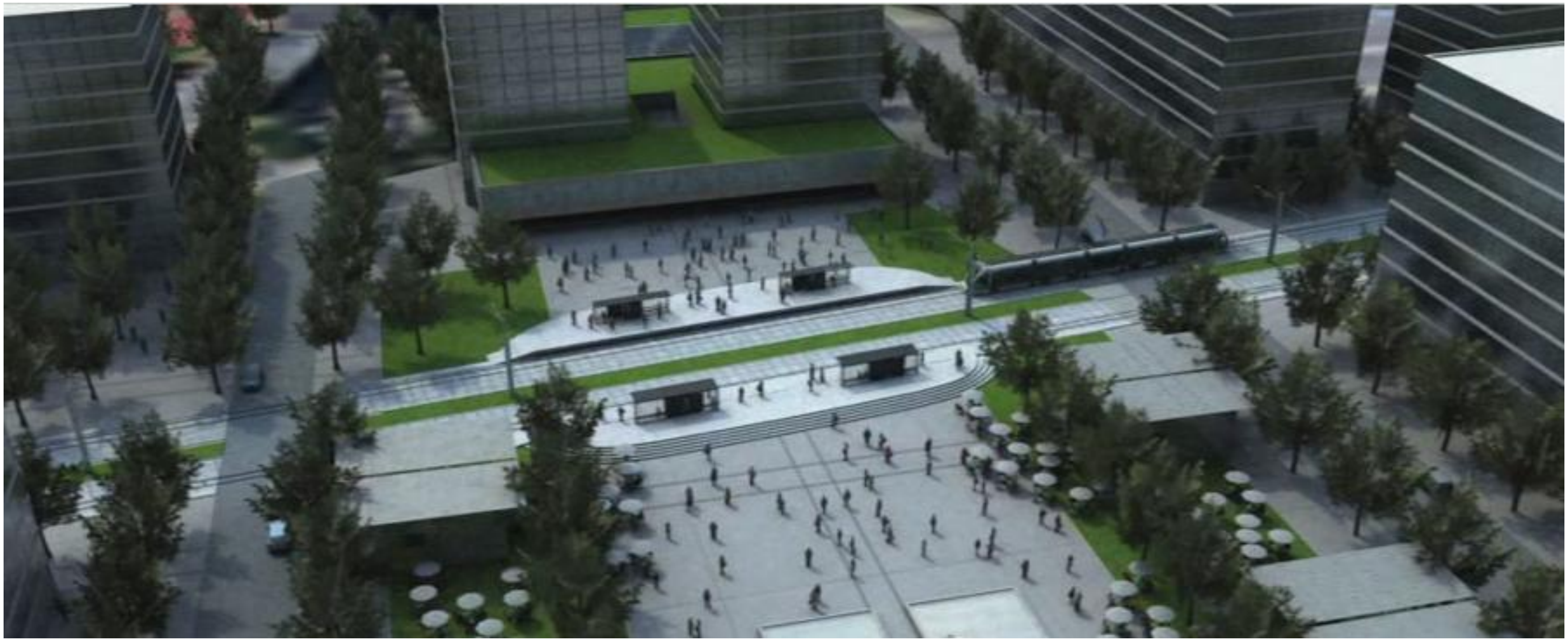
LABORATOIRE d'INFORMATIQUE
de NANTES ATLANTIQUE UMR 6241

TASC SUSTAINS

Interactive Urban Planning with Local Search Techniques: the SUSTAINS Project



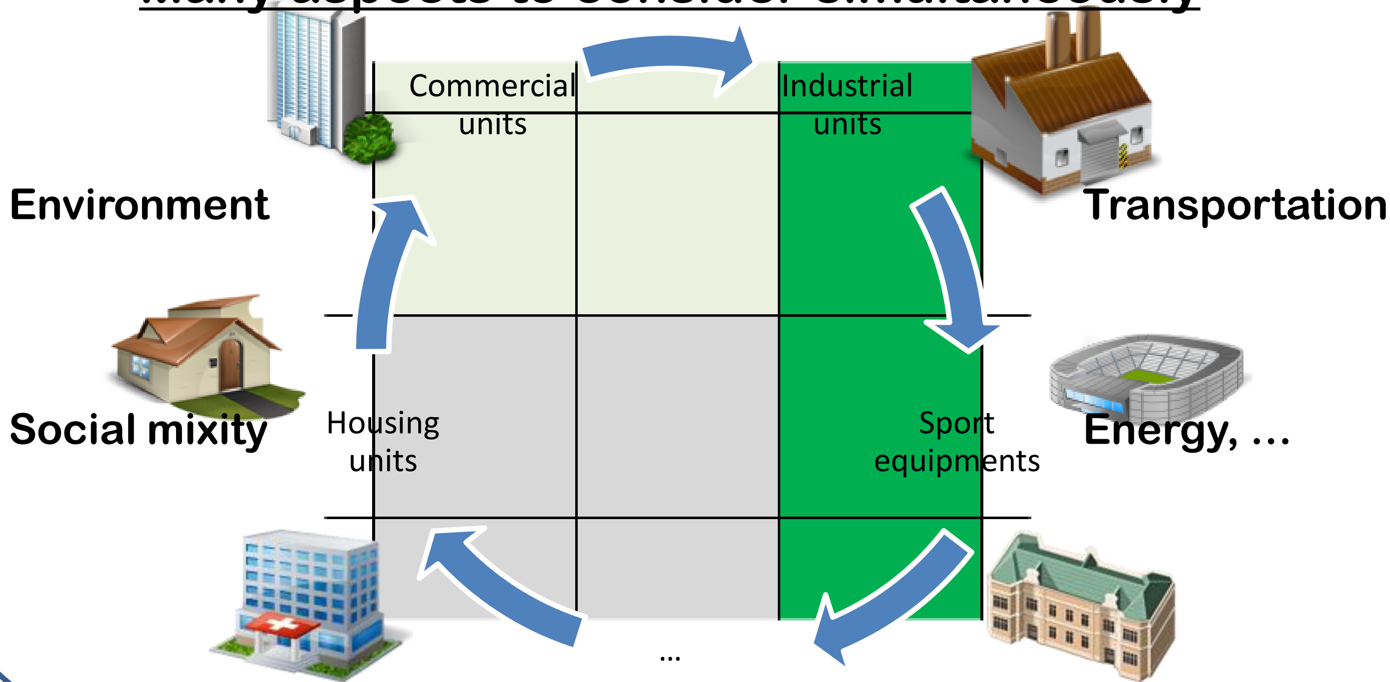
CompSust'12

Combinatorial optimization techniques for sustainable development

By : Bruno BELIN, Marc CHRISTIE and Charlotte TRUCHET.

How to find an optimal layout of urban elements very efficiently ?

Many aspects to consider simultaneously



Outline of the presentation

1. Context
2. Global approach
3. Model and resolution
4. Improvements of the method
5. Conclusion

1

CONTEXT

- Sustainable development
- Stages of the SUSTAINS project
- Our task: Automatic layout of urban elements

Sustainable development

Environmental preservation

Economic viability

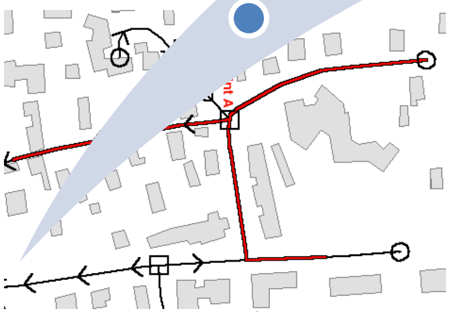
Social equity

Stages of the SUSTAINS project

Interactive communication tool for public engagement

Interactive modification and evaluation of the impacts

Automatic layout of urban elements



Our task: Automatic layout of urban elements

- Concerned only with the first phase of the SUSTAINS project
- Limited to the green field solution for simplicity
- City divided into a grid of regular cells
- Each cell symbolize a city block (80m x 80m)
- How to allocate urban elements to cells ?

2

GLOBAL APPROACH

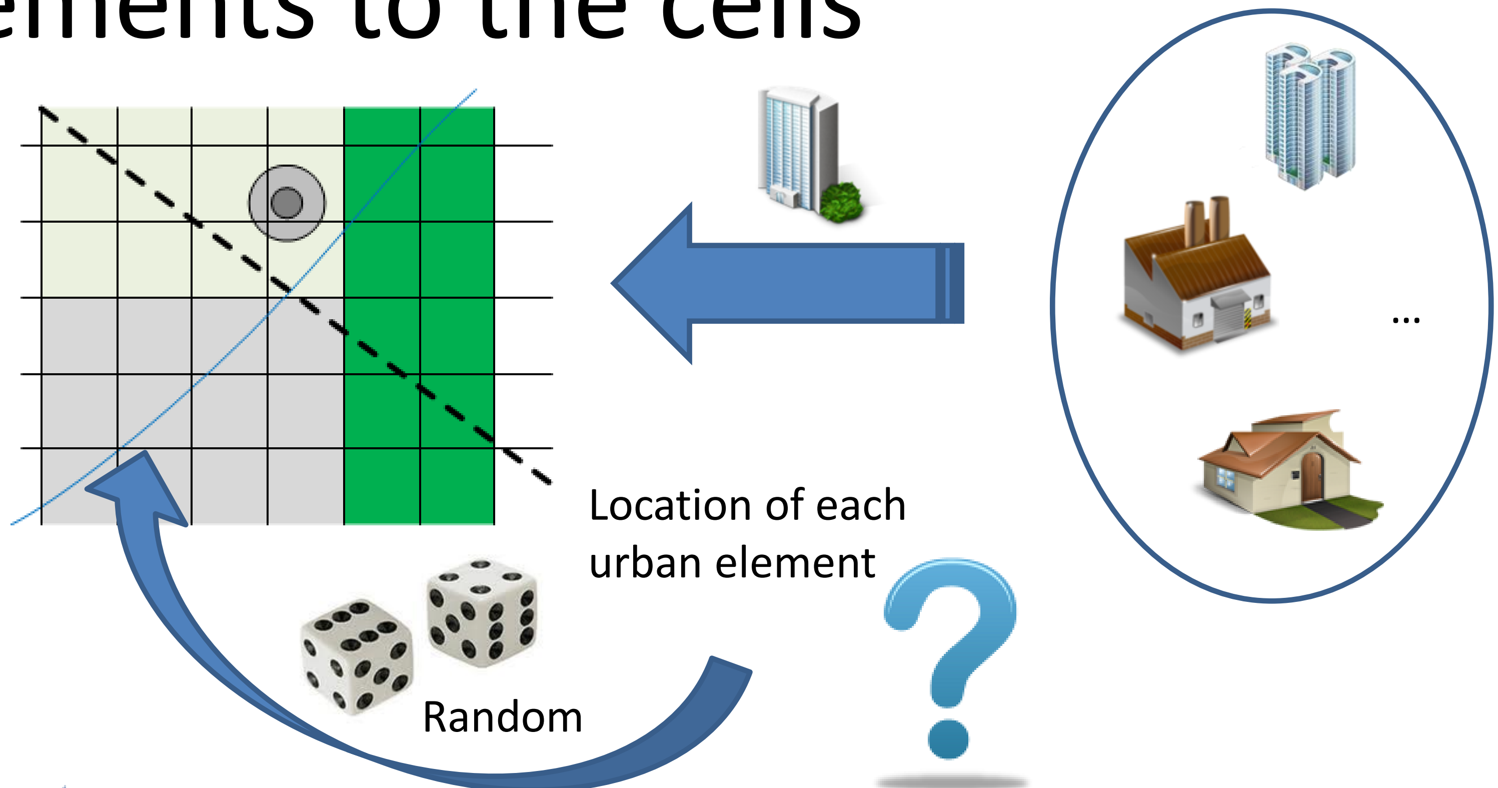
- Sizing to decide the number of city blocks of each urban element
- Producing an initial layout by randomly assigning the urban elements to the cells
- Performing series of swaps between urban elements to maximize a score function

Local Search with a Tabu-like metaheuristic

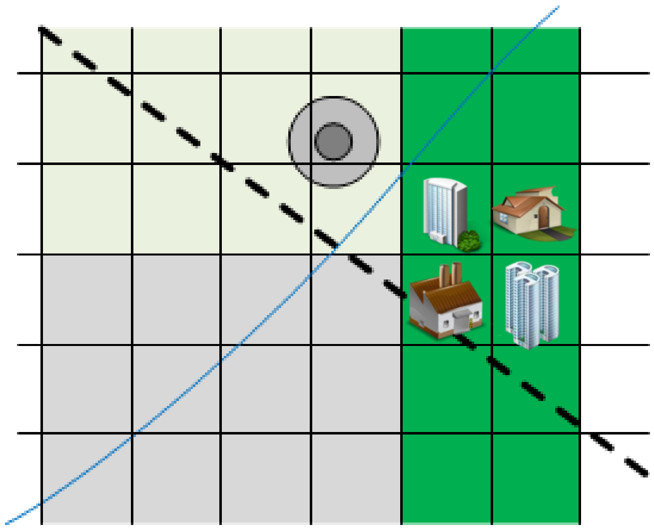
Sizing: to decide the number of city blocks of each urban element

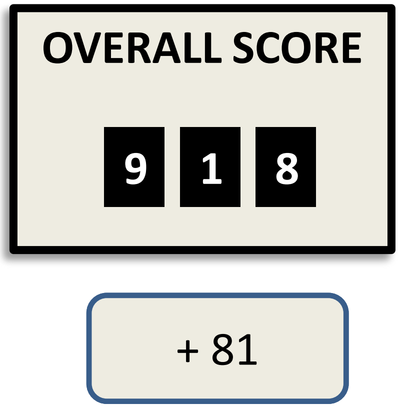
- Incoming data :
  - Population, footprint area, employment rate and country specific indicators
- Outgoing data :
  - Number of schools, residential units, industrial units, ...
- Encode dimensions related to the systemic approach (social mixity, density, ...)

Producing an initial solution by randomly assigning the urban elements to the cells



Performing series of swaps between urban elements to maximize a score function





How to assess the quality of a solution ?

How to solve such a system ?

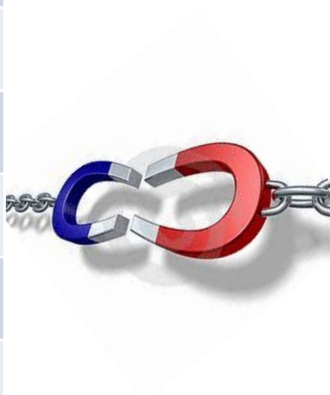
3

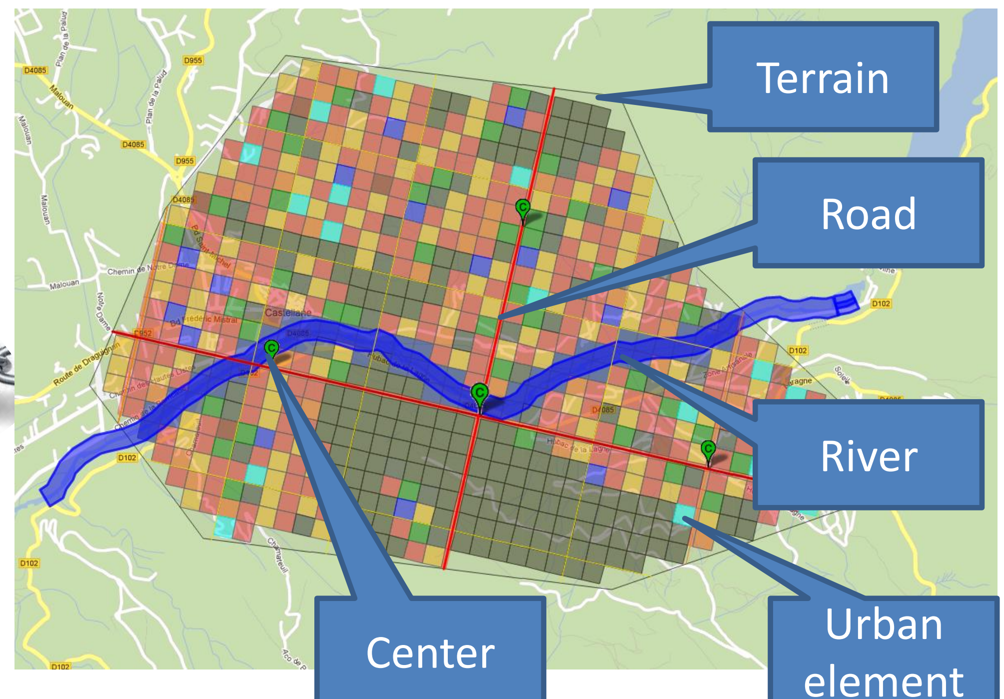
MODEL AND RESOLUTION

How to assess the quality of a solution ?

With attraction-repulsion tables

Urban Elements
Residential units
Industrial units
Commercial units
Schools
Sport equipments
Green spaces
…

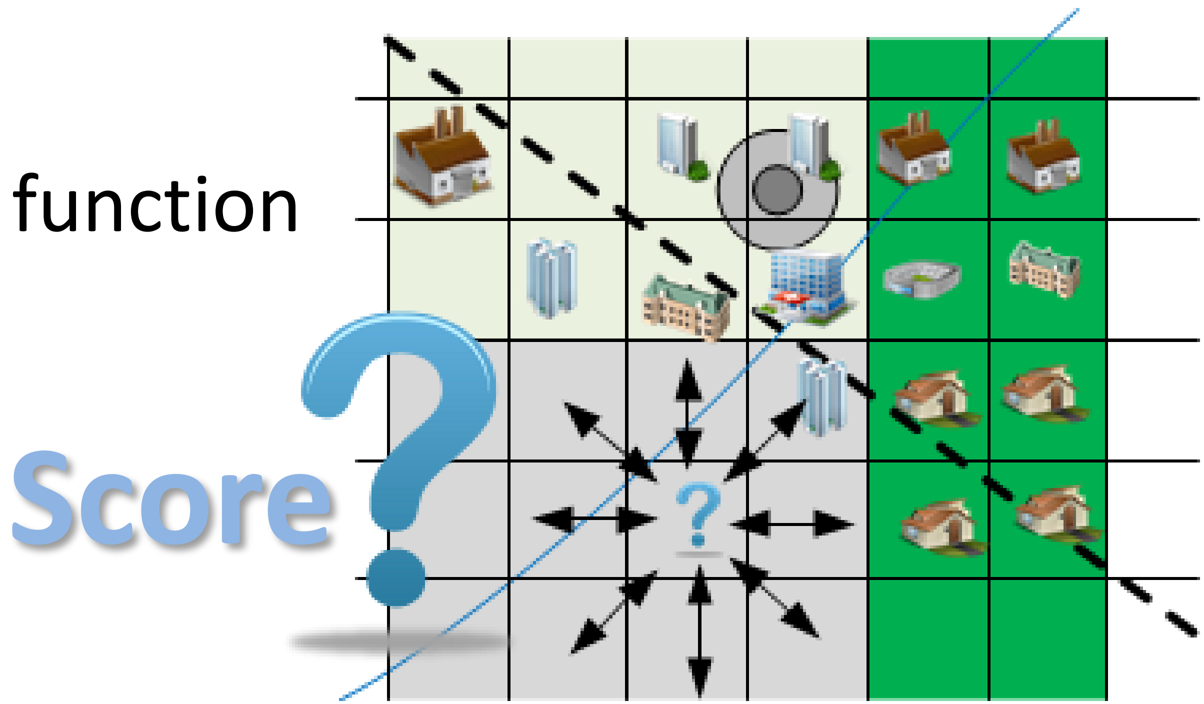




How to assess the quality of a solution ?

And with a score function

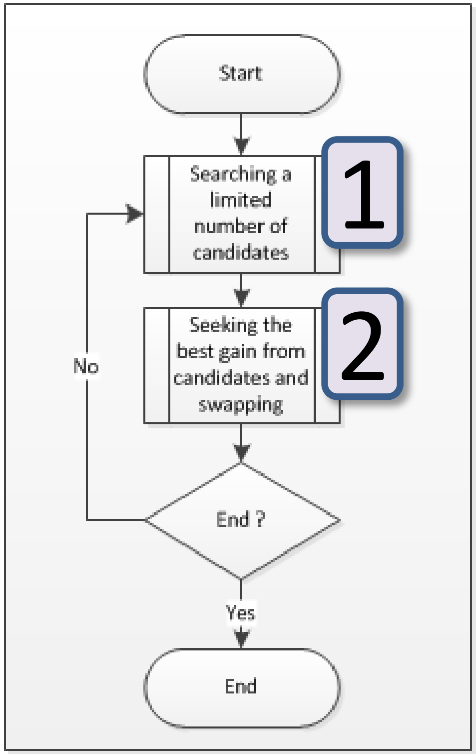
Constraints in action
Terrain
Proximity of a road
Proximity of a river
Proximity of a city center
Proximity of another urban element



- Cell score = sum of scores given by each constraint in action
- Overall score = sum of the scores for each cell
- Best possible configuration = the one which maximizes the overall score

Resolution: General principle

- Searching a limited number of candidates for permutation
- Seeking the best gain from candidates and swapping
- Iterate over first and second stage



Resolution: Searching a limited number of candidates for permutation

1

stage

- Examine the entire grid
- To find a limited number of cells with the lowest score

This cells are potential "candidates" to generate significant gain if swapped with another cell of the grid

Resolution: Seeking the best gain from candidates and swapping

2
stage

- Evaluation of a succession of permutations between urban elements of each candidate and urban elements of the other cells
- Seek the swap which provides the best gain
- Swap the urban elements of these two cells
- Tabu-Like metaheuristic to avoid local maxima

Refinement: Dynamic list of banished candidates

Results

4

IMPROVEMENTS OF THE METHOD

Refinement: Dynamic list of banished candidates

- Candidates unable to produce a benefit by swapping with another cell are inserted in a list of "**banished candidates**"
- Banished candidates will not be considered in subsequent steps
- The oldest banished candidates are removed from the list once it is full.
- If no profit is found: list is reset with a larger size

Results

Basic Local Search: <u>I</u>terative <u>I</u>mprovement

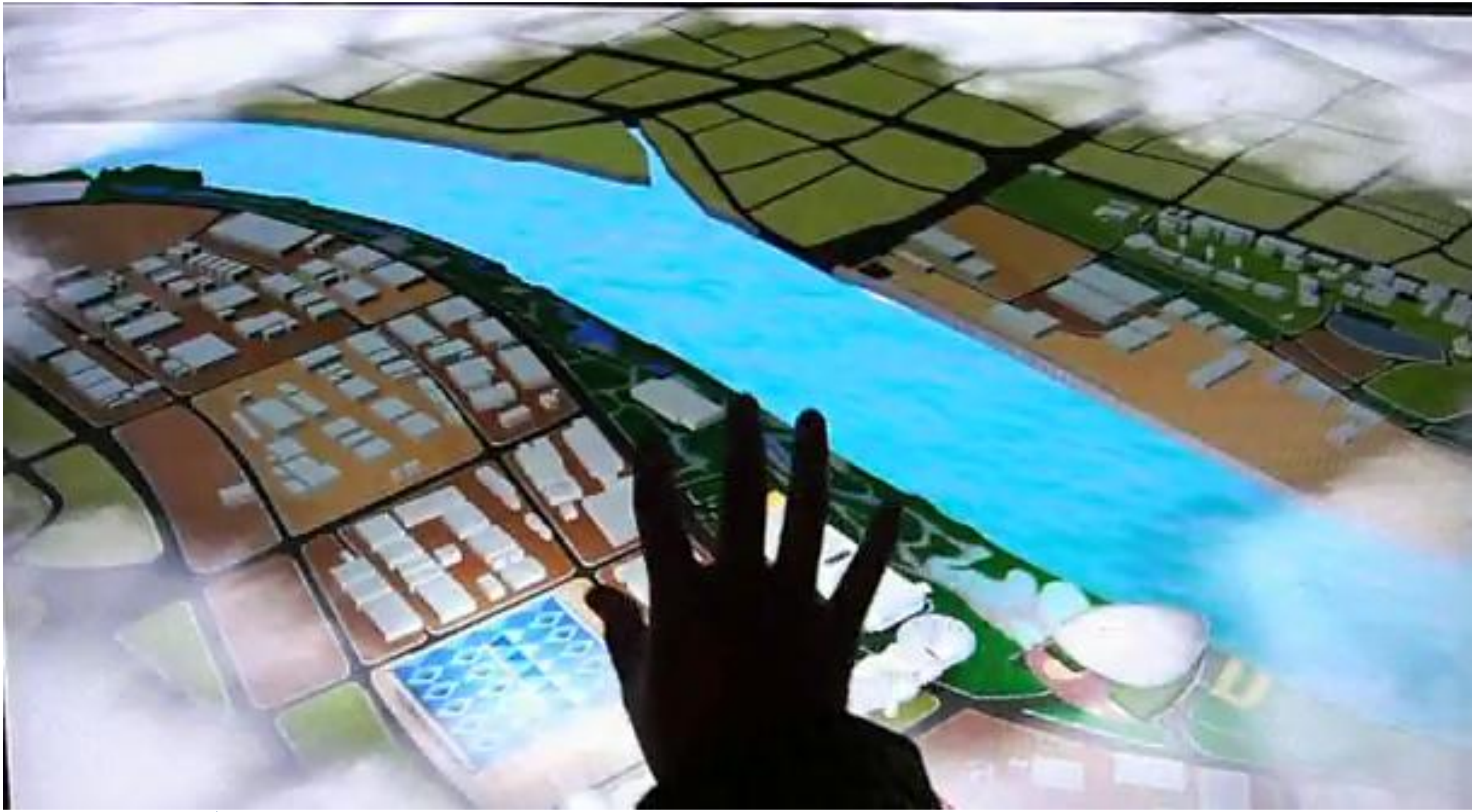
Parameter set	Maximum Candidates	List size of banished candidates	% increase list of banished candidates	Time (sec)
G32-II	20	0	0	45
G32-IIBC	20	100	0	24
G32-IIDBC	20	100	10	**12**
G64-II	100	0	0	1553
G64-IIBC	100	400	0	833
G64-IIDBC	100	400	10	**224**
<u>G32</u> : Grid of 32x32 cells; Distance: 100% indifferent; 3 perturbations of 10 movements				
<u>G64</u> : Grid of 64x64 cells; Distance: 100% distant; No perturbation				

With a grid of 64x64 cells, <u>Dynamic list of banished candidates</u> deliver performance gains up to **seven times**. Almost same scores for all grids.

Toward exciting possibilities

5

CONCLUSION



References

[1] Emile Aarts and Jan K. Lenstra, editors. Local Search in Combinatorial Optimization. John Wiley & Sons, Inc., New York, NY, USA, 1st edition, 1997.

[2] Saskia A. Groenewegen, Ruben M. Smelik, Klaas Jan de Kraker, and Rafael Bidarra. Procedural City Layout Generation Based on Urban Land Use Models. pages 45-48, Munich, Germany, 2009. Proceedings of Eurographics (Short Papers).

[3] Francesca Rossi, Peter van Beek, and Toby Walsh. Handbook of Constraint Programming (Foundations of Articial Intelligence). Elsevier Science Inc., New York, NY, USA, 2006.

Thanks for your attention

Discover more under :

WWW.SUSTAINS.FR